On Average Behaviour of Regular Expressions in Strong Star Normal Form

Sabine Broda, António Machiavelo, Nelma Moreira and Rogério Reis*

CMUP & DM-DCC, Faculdade de Ciências da Universidade do Porto
Rua do Campo Alegre, 4169-007 Porto, Portugal
sbb@dcc.fc.up.pt,ajmachia@fc.up.pt,{nam,rvr}@dcc.fc.up.pt

For regular expressions in (strong) star normal form a large set of efficient algorithms is known, from conversions into finite automata to characterisations of unambiguity. In this paper we study the average complexity of this class of expressions using analytic combinatorics. As it is not always feasible to obtain explicit expressions for the generating functions involved, here we show how to get the required information for the asymptotic estimates with an indirect use of the existence of Puiseux expansions at singularities. We study, asymptotically and on average, the alphabetic size, the size of the ε-follow automaton, and the ratio and the size of these expressions to standard regular expressions.

Keywords. Regular expressions, star normal form, conversions into finite automata, analytic combinatorics, asymptotic average case complexity, Puiseux series.

1. Introduction

A regular expression α is in strong star normal form (ssnf) if for any subexpression of the form $\beta^\star$ or $\beta + \varepsilon$ the language represented by β does not include the empty word, ε. The broader notion of star normal form was introduced by Brüggemann-Klein [7] as a step to improve the construction of the position automaton from a regular expression from cubic to quadratic time. Transforming a regular expression into this normal form can be achieved in linear time, and moreover the position automaton resulting from that normal form coincides with the one of the original expression. In the same paper, the star normal form was also used to characterize certain types of unambiguous expressions. The position automaton construction [11] is a basic conversion between regular expressions and ε-free nondeterministic finite automata (NFA), and several other constructions are known to be its quotients. This is the case for the partial derivative automaton [1, 9] and the follow automaton [16]. Champarnaud et al. [8] showed that if a regular expression is in star normal form and is normalized modulo some regular expression equivalences, the partial derivative automaton is a quotient of the follow automaton. Many other conversions

*This work was partially supported by CMUP (UID/MAT/00144/2013), which is funded by FCT (Portugal) with national (MEC) and European structural funds through the programs FEDER, under the partnership agreement PT2020.

from regular expressions to equivalent NFAs consider automata with transitions labelled by the empty word (ε-NFA). Although the most used of these conversions is the Thompson construction (implemented in many UNIX-like string search commands) [22], an older and more thrifty construction in the use of ε-transitions was presented by Ott and Feinstein in 1961 [19]. An improved version of this construction was redefined by Ilie and Yu, and called the ε-follow automaton. Gulan, Fernau and Gruber [14, 12, 13] studied the optimal (worst-case) size for all known constructions from regular expressions to ε-NFAs. It turns out that the optimal construction corresponds to the conversion of a regular expression in strong star normal form into an ε-follow automaton.

All this motivated us to study the average-case complexity of regular expressions in strong star normal form, as well as their conversions to NFAs. In previous work, we studied the asymptotic average complexity for some of the above mentioned conversions from regular expressions using the framework of analytic combinatorics [2, 4, 5], which relates the enumeration of combinatorial objects to the algebraic and complex analytic properties of generating functions. In particular, generating functions can be seen as complex analytic functions, and the study of their behaviour around their dominant singularities gives access to the asymptotic form of their coefficients. Starting with an unambiguous grammar for the set of regular expressions over a given alphabet, and a non-negative measure, the symbolic method allows to obtain a generating function associated with the sequence of the (finite) number of expressions of measure value n. Multivariate generating functions can be used to analyse different measures apart from the size of combinatorial objects, e.g. the number of states of the automaton resulting from a given conversion method applied to a regular expression of given size, and thus allow to obtain estimates for the average values of those measures.

While in previous work we were able to get explicit expressions for the generating functions involved, here that would be unmanageable. Using the existence of a Puiseux expansion at a singularity, we show how to get the required information for the asymptotic estimates from an algebraic equation satisfied by the generating function, without actually computing that expansion. We note that the technique here presented allows to find, for the combinatorial classes considered, the form of the function without knowing beforehand the explicit value of the singularity. This provides a very useful method, at least for some combinatorial classes, that circumvents some of the more cumbersome steps of the *Algebraic Coefficient Asymptotics* algorithm presented by Flajolet and Sedgewick [10], pages $504 - 505$, as well as the need to know *a priori* the type of the singularity.

We use this method to derive the asymptotic estimates for the number of regular expressions in ssnf of a given size, as well as a parametric function of several related measures, which can give us, in particular, the alphabetic size of the expressions or the size of the ε-follow automaton, on average. We note that this parametric function cannot be used to estimate the average size of the position automaton construction for regular expressions in ssnf, but it is possible to obtain a system of

equations satisfied by the generating functions associated with this measure.

A preliminary version of this paper was presented in [6]. In the next section, we review some basics on regular expressions and NFAs. In Section 3, we consider the transformation into strong star normal form and give some characterisations of expressions in this form. Section 4 describes a shortcut to obtain asymptotic estimates of the coefficients of generating functions. This is used in Section 5 to obtain the estimates mentioned above. Applying the same technique, in Section 6 we estimate the asymptotic ratio between a general expression and the corresponding expression in ssnf. Some experiments corroborating those estimates are presented in Section 7. Conclusions are drawn in Section 8.

2. Regular Expressions and NFAs

We consider the grammar for regular expressions proposed by Gruber and Gulan in [12, 13], which has the major advantage of avoiding many redundant expressions built with the symbols ε and $\emptyset$. Given an alphabet $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$ of size k, the set $\mathcal{R}_k$ of *regular expressions*, α, over Σ is defined by the following grammar,

$$\begin{aligned} &\alpha := \emptyset \mid \varepsilon \mid \beta, \\ &\beta := \sigma_1 \mid \cdots \mid \sigma_k \mid (\beta + \beta) \mid (\beta \cdot \beta) \mid \beta^\star \mid \beta^? \end{aligned} \tag{1}$$

where the operator $\cdot$ (concatenation) is often omitted. The language associated with α is denoted by $\mathcal{L}(\alpha)$ and is defined as usual, with $\mathcal{L}(\beta^?) = \mathcal{L}(\beta) \cup \{\varepsilon\}$. It is clear that $\beta^?$ is equivalent to the standard regular expression $\beta + \varepsilon$.

For the *size* of a regular expression α, denoted by $|\alpha|$, we will consider reverse polish notation length, i.e., the number of symbols in α, not counting parentheses. The number of letters in α is denoted by $|\alpha|_\Sigma$, and usually called *alphabetic size*. The number of occurrences of each operator $c \in \{+, \cdot, \star, ?\}$ is denoted by $|\alpha|_c$. One has

$$|\alpha| = |\alpha|_\Sigma + |\alpha|_+ + |\alpha|_\cdot + |\alpha|_\star + |\alpha|_?.$$

A *nondeterministic finite automaton* is a tuple $\mathcal{N} = \langle Q, \Sigma, \delta, q_0, F\rangle$, where Q is a finite set of states, Σ is the alphabet, $\delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$ is the transition relation, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states. The *size* of an NFA $\mathcal{N}$ is $|\mathcal{N}| = |\mathcal{N}|_Q + |\mathcal{N}|_\delta$, where $|\mathcal{N}|_Q = |Q|$ and $|\mathcal{N}|_\delta = |\delta|$. An NFA that has transitions labelled with ε is an ε-NFA. The *language* accepted by an automaton $\mathcal{N}$ is $\mathcal{L}(\mathcal{N}) = \{\ w \in \Sigma^\star \mid \delta(q_0, w) \cap F \neq \emptyset\ \}$, where δ is naturally extended to sets of states and words.

Conversion of a regular expression into an equivalent NFA can be defined by induction on the structure of the regular expression. Let $\mathcal{N}_\alpha$ denote the automaton corresponding to a regular expression α. In Figure 1 we present the construction of the ε-follow automaton, $\mathcal{A}_{\varepsilon\mathsf{f}}(\beta)$ [16]. The size of the $\mathcal{A}_{\varepsilon\mathsf{f}}(\beta)$ for the atomic expressions $\emptyset$, ε, and $\sigma \in \Sigma$ is 2, 3 and 3, respectively. For the remaining constructions, the size of the resulting automaton equals the sum of the

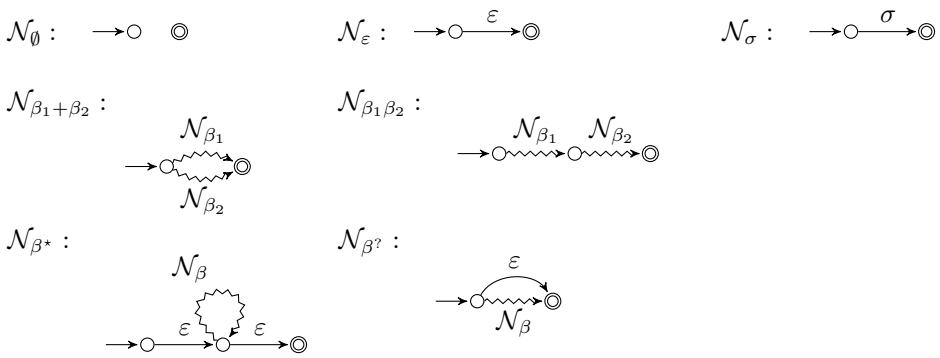

Fig. 1. The ε-follow construction, $\mathcal{A}_{\varepsilon\mathsf{f}}$.

sizes of its constituents plus some constant. For instance, for the operator + one has $|\mathcal{N}_{\beta_1+\beta_2}|_Q = |\mathcal{N}_{\beta_1}|_Q + |\mathcal{N}_{\beta_2}|_Q - 2$, $|\mathcal{N}_{\beta_1+\beta_2}|_\delta = |\mathcal{N}_{\beta_1}|_\delta + |\mathcal{N}_{\beta_2}|_\delta$, and thus $|\mathcal{N}_{\beta_1+\beta_2}| = |\mathcal{N}_{\beta_1}| + |\mathcal{N}_{\beta_2}| - 2$. This can be generalized by considering constants $(c_\emptyset, c_\varepsilon, c_\sigma, c_+, c_\bullet, c_\star, c_?)$ that define functions that can be used to compute several interesting measures. For example, using $(2, 2, 2, -2, -1, 1, 0)$ one gets the number of states; the number of transitions are computed using $(0, 1, 1, 0, 0, 2, 1)$, and the combined size corresponds to $(2, 3, 3, -2, -1, 3, 1)$.

We note that the worst-case complexity for this conversion can be reached for expressions with only one letter and $n-1$ stars. For such an expression of size n, the corresponding $\mathcal{A}_{\varepsilon\mathsf{f}}$ automaton has size $3n$. Broda et al. [4] presented an asymptotic estimate of the average size of $\mathcal{A}_{\varepsilon\mathsf{f}}$ and showed that its limit is $\frac{3}{4}n$ as k goes to ∞.

The position or Glushkov automaton from a regular expression α, $\mathcal{A}_{\mathsf{pos}}(\alpha)$, can also be defined inductively on the structure of α [11, 23, 21]. However, its size can not be expressed as a sum of the sizes of its constituents plus a fixed constant. We define the set of positions of α by $\mathsf{Pos}(\alpha) = \{1, \ldots, |\alpha|_\Sigma\}$. Let $\overline{\alpha}$ denote the *marked expression* obtained from α by indexing each letter with its position in α Let the sets *first*, *last* and *follow* be $\mathsf{First}(\alpha) = \{\ i \mid \sigma^i w \in \mathcal{L}(\overline{\alpha})\ \}$, $\mathsf{Last}(\alpha) = \{\ i \mid w\sigma^i \in \mathcal{L}(\overline{\alpha})\ \}$ and $\mathsf{Follow}(\alpha, i) = \{\ j \mid u\sigma^i\sigma^j v \in \mathcal{L}(\overline{\alpha})\ \}$, respectively. The *Glushkov automaton* for α is $\mathcal{A}_{\mathsf{pos}}(\alpha) = (\mathsf{Pos}(\alpha) \cup \{0\}, \Sigma, \delta_{\mathsf{pos}}, 0, F)$, with $\delta_{\mathsf{pos}} = \{\ (0, \overline{\sigma}^j, j) \mid j \in \mathsf{First}(\alpha)\ \} \cup \{\ (i, \overline{\sigma}^j, j) \mid\ j \in \mathsf{Follow}(\alpha, i)\ \}$ and $F = \mathsf{Last}(\alpha) \cup \{0\}$ if $\varepsilon \in \mathcal{L}(\alpha)$, and $F = \mathsf{Last}(\alpha)$, otherwise. The number of states is the alphabetic size of α plus one and the number of transitions is the sum of the cardinalities of the sets *first* and *follow*. Thus, the size of $\mathcal{A}_{\mathsf{pos}}(\alpha)$ can be calculated using inductive definitions of the sets *first*, *last* and *follow*. In the worst case the size of $\mathcal{A}_{\mathsf{pos}}(\alpha)$ is quadratic in the size of α. It is known that asymptotically and on average for an expression α of size n its alphabetic size is $\frac{n}{2}$ (thus the number of states of $\mathcal{A}_{\mathsf{pos}}$) and $|\mathcal{A}_{\mathsf{pos}}(\alpha)|$ is $O(n)$ [18, 3].

3. Strong Star Normal Form

Brüggemann-Klein [7] introduced the notion of *star normal form* of a regular expression and defined a function that given a regular expression computes an equivalent star normal form. Gulan and Gruber simplified that definition and adapted it for regular expressions with the operator $^?$. The result $\mathsf{ssnf}(\alpha)$ was called the *strong star normal form* of α. To define the function ssnf we denote by β_ε and $\beta_{\overline{\varepsilon}}$ regular expressions whose language includes ε or not, respectively.

Definition 1. *The operator* ssnf *is inductively defined as follows.*

$$\begin{aligned}
\mathsf{ssnf}(\emptyset) &= \emptyset \\
\mathsf{ssnf}(\varepsilon) &= \varepsilon \\
\mathsf{ssnf}(\sigma) &= \sigma \\
\mathsf{ssnf}(\beta_1 + \beta_2) &= \mathsf{ssnf}(\beta_1) + \mathsf{ssnf}(\beta_2) \\
\mathsf{ssnf}(\beta_1\beta_2) &= \mathsf{ssnf}(\beta_1) \cdot \mathsf{ssnf}(\beta_2) \\
\mathsf{ssnf}(\beta^\star) &= \mathit{ss}(\beta)^\star \\
\mathsf{ssnf}(\beta_\varepsilon^?) &= \mathsf{ssnf}(\beta_\varepsilon) \\
\mathsf{ssnf}(\beta_{\overline{\varepsilon}}^?) &= \mathsf{ssnf}(\beta_{\overline{\varepsilon}})^?,
\end{aligned}$$

where

$$\begin{aligned}
\mathit{ss}(\emptyset) = \mathit{ss}(\varepsilon) &= \emptyset \\
\mathit{ss}(\sigma) &= \sigma \\
\mathit{ss}(\beta_1 + \beta_2) &= \mathit{ss}(\beta_1) + \mathit{ss}(\beta_2) \\
\mathit{ss}(\beta_\varepsilon\beta'_\varepsilon) &= \mathit{ss}(\beta_\varepsilon) + \mathit{ss}(\beta'_\varepsilon) \\
\mathit{ss}(\beta_\varepsilon\beta_{\overline{\varepsilon}}) &= \mathsf{ssnf}(\beta_\varepsilon) \cdot \mathsf{ssnf}(\beta_{\overline{\varepsilon}}) \\
\mathit{ss}(\beta_{\overline{\varepsilon}}\beta) &= \mathsf{ssnf}(\beta_{\overline{\varepsilon}}) \cdot \mathsf{ssnf}(\beta) \\
\mathit{ss}(\beta^\star) &= \mathit{ss}(\beta) \\
\mathit{ss}(\beta^?) &= \mathit{ss}(\beta).
\end{aligned}$$

An expression α *is in* strong star normal form *if* $\alpha = \mathsf{ssnf}(\alpha)$.

For a regular expression α, one has $\mathcal{L}(\mathsf{ssnf}(\alpha)) = \mathcal{L}(\alpha)$ and $|\mathsf{ssnf}(\alpha)| \leq |\alpha|$. The following theorem characterises the regular expressions in strong star normal form.

Theorem 2. [13] *A regular expression* α *is in strong star normal form, i.e.* $\alpha = \mathsf{ssnf}(\alpha)$*, if and only if for every subexpression* $\beta^\star$ *or* $\beta^?$ *of* α*, one has* $\varepsilon \notin \mathcal{L}(\beta)$.

Using this theorem it is possible to write a context-free grammar for regular expressions in ssnf, i.e. in which every subexpression of the form $\beta^\star$ or $\beta^?$ satisfies

$\varepsilon \notin \mathcal{L}(\beta)$. The set $\mathcal{S}_k$ of *regular expressions in* ssnf over Σ is defined by:

$$\begin{aligned} \alpha &:= \varepsilon \mid \emptyset \mid \beta_\varepsilon \mid \beta_{\overline{\varepsilon}} \\ \beta_\varepsilon &:= \beta_\varepsilon\beta_\varepsilon \mid \beta_\varepsilon + \beta_{\overline{\varepsilon}} \mid \beta_{\overline{\varepsilon}} + \beta_\varepsilon \mid \beta_\varepsilon + \beta_\varepsilon \mid \beta_{\overline{\varepsilon}}^\star \mid \beta_{\overline{\varepsilon}}^? \\ \beta_{\overline{\varepsilon}} &:= \sigma_1 \mid \cdots \mid \sigma_k \mid \beta_{\overline{\varepsilon}}\beta_{\overline{\varepsilon}} \mid \beta_{\overline{\varepsilon}}\beta_\varepsilon \mid \beta_\varepsilon\beta_{\overline{\varepsilon}} \mid \beta_{\overline{\varepsilon}} + \beta_{\overline{\varepsilon}}, \end{aligned} \tag{2}$$

where β_ε are regular expressions whose languages include ε, while for $\beta_{\overline{\varepsilon}}$, $\varepsilon \notin \mathcal{L}(\beta_{\overline{\varepsilon}})$. In the remaining of the paper we will use β to denote either of these expressions.

The following theorem summarises the results by Gruber and Gulan [12] (see also Gulan [13]) on the size of $\mathcal{A}_{\varepsilon\mathsf{f}}$, which show that for expressions in ssnf the worst-case is about half the size as for general expressions ($\frac{22}{15} \simeq \frac{3}{2}$).

Theorem 3. *Let α be in* ssnf *of size n and alphabetic size m. Then, $\mathcal{A}_{\varepsilon\mathsf{f}}(\alpha)$ has size at most* $\min(\frac{22}{15}(n+1)+1, \frac{22}{5}m+1)$.

As already stated in the introduction, if a regular expression is in ssnf the position automaton can be computed in quadratic time [7, 3]. This is due to a property of the computation of the *follow* sets (Lemma 4), which is easily shown using the grammar for expressions in ssnf given above. We give inductive definitions of the sets *first*, *last* and *follow* for this kind of expressions. Given a marked expression following grammar (2), we have

$$\begin{aligned} \mathsf{First}(\varepsilon) = \mathsf{First}(\emptyset) &= \emptyset, \\ \mathsf{First}(\sigma_i) &= \{i\}, \\ \mathsf{First}(\beta_1 + \beta_2) &= \mathsf{First}(\beta_1) \cup \mathsf{First}(\beta_2), \\ \mathsf{First}(\beta_\varepsilon\beta) &= \mathsf{First}(\beta_\varepsilon) \cup \mathsf{First}(\beta), \\ \mathsf{First}(\beta_{\overline{\varepsilon}}\beta) &= \mathsf{First}(\beta_{\overline{\varepsilon}}), \\ \mathsf{First}(\beta_{\overline{\varepsilon}}^\star) &= \mathsf{First}(\beta_{\overline{\varepsilon}}), \\ \mathsf{First}(\beta_{\overline{\varepsilon}}^?) &= \mathsf{First}(\beta_{\overline{\varepsilon}}). \end{aligned}$$

The definition of the set $\mathsf{Last}(\alpha)$ coincides with the one of $\mathsf{First}(\alpha)$ except for concatenation, where it is given by:

$$\begin{aligned} \mathsf{Last}(\beta\beta_\varepsilon) &= \mathsf{Last}(\beta_\varepsilon) \cup \mathsf{Last}(\beta), \\ \mathsf{Last}(\beta\beta_{\overline{\varepsilon}}) &= \mathsf{Last}(\beta_{\overline{\varepsilon}}). \end{aligned}$$

Finally, for the set *follow* we have

$$\begin{aligned} \mathsf{Follow}(\varepsilon) = \mathsf{Follow}(\emptyset) = \mathsf{Follow}(\sigma) &= \emptyset, \\ \mathsf{Follow}(\beta_1 + \beta_2) &= \mathsf{Follow}(\beta_1) \cup \mathsf{Follow}(\beta_2), \\ \mathsf{Follow}(\beta_1\beta_2) &= \mathsf{Follow}(\beta_1) \cup \mathsf{Follow}(\beta_2) \cup \mathsf{Last}(\beta_1) \times \mathsf{First}(\beta_2), \\ \mathsf{Follow}(\beta_{\overline{\varepsilon}}^\star) &= \mathsf{Follow}(\beta_{\overline{\varepsilon}}) \cup \mathsf{Last}(\beta_{\overline{\varepsilon}}) \times \mathsf{First}(\beta_{\overline{\varepsilon}}), \\ \mathsf{Follow}(\beta_{\overline{\varepsilon}}^?) &= \mathsf{Follow}(\beta_{\overline{\varepsilon}}). \end{aligned}$$

Except for the union in the equation for the * operator, it is obvious that all other unions occurring in these definitions are disjoint. Next, we show that this is also the case for that equation.

Lemma 4. $\mathsf{Follow}(\beta_{\overline{\varepsilon}}) \cap (\mathsf{Last}(\beta_{\overline{\varepsilon}}) \times \mathsf{First}(\beta_{\overline{\varepsilon}})) = \emptyset$.

Proof. By induction, considering the production rules for $\beta_{\overline{\varepsilon}}$. Given an expression $\beta_{\overline{\varepsilon}}$, let $\mathsf{FLF}(\beta_{\overline{\varepsilon}})$ denote the expression $\mathsf{Follow}(\beta_{\overline{\varepsilon}}) \cap \mathsf{Last}(\beta_{\overline{\varepsilon}}) \times \mathsf{First}(\beta_{\overline{\varepsilon}})$.

$$
\begin{aligned}
\mathsf{FLF}(\sigma^i) &= \emptyset \\
\mathsf{FLF}(\beta_{\overline{\varepsilon}}\beta'_{\overline{\varepsilon}}) &= (\mathsf{Follow}(\beta_{\overline{\varepsilon}}) \cup \mathsf{Follow}(\beta'_{\overline{\varepsilon}}) \cup \mathsf{Last}(\beta_{\overline{\varepsilon}}) \times \mathsf{First}(\beta'_{\overline{\varepsilon}})) \\
&\quad \cap \mathsf{Last}(\beta'_{\overline{\varepsilon}}) \times \mathsf{First}(\beta_{\overline{\varepsilon}}) = \emptyset \\
\mathsf{FLF}(\beta_{\overline{\varepsilon}}\beta_{\varepsilon}) &= (\mathsf{Follow}(\beta_{\overline{\varepsilon}}) \cup \mathsf{Follow}(\beta_{\varepsilon}) \cup \mathsf{Last}(\beta_{\overline{\varepsilon}}) \times \mathsf{First}(\beta_{\varepsilon})) \\
&\quad \cap (\mathsf{Last}(\beta_{\overline{\varepsilon}}) \times \mathsf{First}(\beta_{\overline{\varepsilon}}) \cup \mathsf{Last}(\beta_{\varepsilon}) \times \mathsf{First}(\beta_{\overline{\varepsilon}})) = \emptyset \\
\mathsf{FLF}(\beta_{\varepsilon}\beta_{\overline{\varepsilon}}) &= (\mathsf{Follow}(\beta_{\varepsilon}) \cup \mathsf{Follow}(\beta_{\overline{\varepsilon}}) \cup \mathsf{Last}(\beta_{\varepsilon}) \times \mathsf{First}(\beta_{\overline{\varepsilon}})) \\
&\quad \cap (\mathsf{Last}(\beta_{\overline{\varepsilon}}) \times \mathsf{First}(\beta_{\varepsilon}) \cup \mathsf{Last}(\beta_{\overline{\varepsilon}}) \times \mathsf{First}(\beta_{\overline{\varepsilon}})) = \emptyset \\
\mathsf{FLF}(\beta_{\overline{\varepsilon}} + \beta'_{\overline{\varepsilon}}) &= (\mathsf{Follow}(\beta_{\overline{\varepsilon}}) \cup \mathsf{Follow}(\beta'_{\overline{\varepsilon}})) \\
&\quad \cap (\mathsf{Last}(\beta_{\overline{\varepsilon}}) \times \mathsf{First}(\beta_{\overline{\varepsilon}}) \cup \ \mathsf{Last}(\beta_{\overline{\varepsilon}}) \times \mathsf{First}(\beta'_{\overline{\varepsilon}}) \\
&\qquad \cup \mathsf{Last}(\beta'_{\overline{\varepsilon}}) \times \mathsf{First}(\beta_{\overline{\varepsilon}}) \\
&\qquad \cup \mathsf{Last}(\beta'_{\overline{\varepsilon}}) \times \mathsf{First}(\beta'_{\overline{\varepsilon}})) = \emptyset.
\end{aligned}
$$

Note that, only the last equation uses the induction hypothesis. □

Using Lemma 4, it is immediate that $\mathcal{A}_{\mathsf{pos}}$ can be computed in $O(n^2)$, which provides an alternative proof of this result given by Brüggemann-Klein [7].

4. Asymptotic Average Complexity

Let $A(z) = \sum_n a_n z^n$ be the generating function associated with some combinatorial class $\mathcal{A}$ (*cf.* [10]). Given some measure of the objects of the class, the coefficient a_n represents the sum of the values of this measure for all objects of size n. We will use the notation $[z^n]A(z)$ for a_n. The generating function $A(z)$ can be seen as a complex analytic function, and the study of its behaviour around its dominant singularity ρ (when unique) gives us access to the asymptotic form of its coefficients. In particular, if $A(z)$ is analytic in some indented disc neighbourhood of ρ, then one has the following [10, 4]:

(1) if $A(z) = a - b\sqrt{1 - z/\rho} + o\left(\sqrt{1 - z/\rho}\right)$, with $a, b \in \mathbb{R}$, $b \neq 0$, then

$$[z^n]A(z) \sim \frac{b}{2\sqrt{\pi}}\, \rho^{-n} n^{-3/2}; \tag{3}$$

(2) if $A(z) = \frac{c}{\sqrt{1-z/\rho}} + o\left(\frac{1}{\sqrt{1-z/\rho}}\right)$, with $c \in \mathbb{R}^*$, then

$$[z^n]A(z) \sim \frac{c}{\sqrt{\pi}}\,\rho^{-n} n^{-1/2}. \tag{4}$$

Applying this result for the generating function $R_k(z)$, corresponding to the number of expressions in $\mathcal{R}_k$ of size n, the following asymptotic values were obtained in Broda et al. [4]:

$$[z^n]R_k(z) \sim \frac{\sqrt[4]{2k}\sqrt{\rho_k}}{4\sqrt{\pi}}\rho_k^{-(n+1)}(n+1)^{-3/2}, \text{ with } \rho_k = \frac{1}{2(\sqrt{2k}+1)}. \tag{5}$$

In the same paper, the average size of the ε-follow automata construction was studied, and it was shown that, as the alphabet grows, the size of $\mathcal{A}_{\varepsilon\mathsf{f}}$ approaches $0.75n$, asymptotically and on average.

Let us now give a generic description of the method used for the combinatorial classes that show up within the present paper. From a grammar one obtains, by the symbolic method expounded in [10], a set of polynomial equations involving the generating function of whose coefficients we want to have an asymptotic estimate. Computing a Gröbner basis for the ideal generated by those polynomials, one gets an algebraic equation for that generating function $w = w(z)$, i.e., an equation of the form

$$G(z, w) = 0,$$

where $G(z, w)$ is a polynomial in $\mathbb{Z}[z][w]$, of which $w(z)$ is a root.

Since $w(z)$ is the generating function of a combinatorial class, thus a series with non-negative integer coefficients, which is not a polynomial, it must have, by Pringsheim's Theorem (*cf.* [10], Thm IV.6), a real positive singularity, ρ, smaller than or equal to 1. In all that follows we will assume that there is no other singularity with that norm, which is the case of all generating functions dealt with in this paper, as we will see. At this singularity ρ two cases may occur: either $\lim_{z\to\rho} w(z) = a$, a positive real number, or $\lim_{z\to\rho} w(z) = +\infty$.

In the first case, after making the change of variable $s = 1 - z/\rho$, one knows that $w = w(s)$ has a Puiseux series expansion at the singularity $s = 0$, i.e., there exists a slit neighbourhood of that point in which $w(s)$ has a representation as a power series with fractional powers (*cf.* [15], Chap. 12). In particular, w must have the form

$$w(s) = a - g(s)s^{\alpha}, \tag{6}$$

for some $a \in \mathbb{R}$, $\alpha \in \mathbb{Q}^+$, the first positive exponent of that expansion, and $g(s)$ such that $g(s) = b + h(s)s^{\beta}$, $h(0) \neq 0$, $\beta \in \mathbb{Q}^+$, and $b \in \mathbb{R}^*$. We will show that, under some generic conditions that happen to be satisfied in all the cases treated below, one has $\alpha = \frac{1}{2}$ or $\alpha = -\frac{1}{2}$. One then needs to find the values of ρ and of b or c, depending on the case, to use either (3) or (4) to obtain the sought-after asymptotic estimates of the coefficients of $w(z)$.

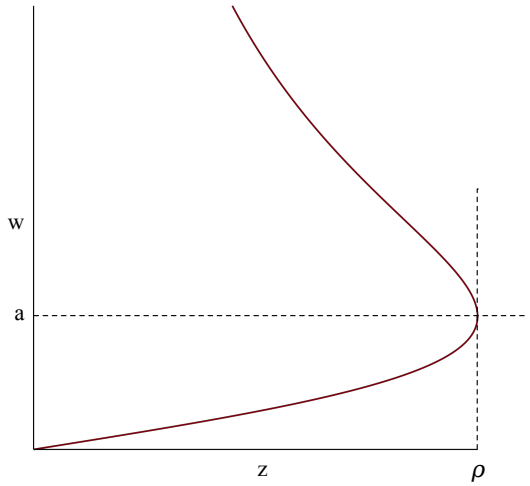

Fig. 2. Generic shape of $G(z, w)$ near its dominant singularity.

Using Taylor expansion of $G(z, w)$ at (ρ, a),

$$\begin{aligned} G(z,w) =& G(\rho,a) + \frac{\partial G}{\partial z}(\rho,a)(z-\rho) + \frac{\partial G}{\partial w}(\rho,a)(w-a) + \\ &+ \frac{1}{2}\frac{\partial^2 G}{\partial z^2}(\rho,a)(z-\rho)^2 + \frac{1}{2}\frac{\partial^2 G}{\partial w^2}(\rho,a)(w-a)^2 + \\ &+ \frac{\partial^2 G}{\partial z\,\partial w}(\rho,a)(z-\rho)(w-a) + \cdots, \end{aligned}$$

and noticing that $G(z, w(z)) = 0$, that $G(\rho, a) = 0$, and using Equation (6), one has,

$$\begin{aligned} 0 =& -\frac{\partial G}{\partial z}(\rho,a)\rho s - \frac{\partial G}{\partial w}(\rho,a)g(s)s^{\alpha} + \frac{1}{2}\frac{\partial^2 G}{\partial z^2}(\rho,a)\rho^2 s^2 + \\ &+ \frac{1}{2}\frac{\partial^2 G}{\partial w^2}(\rho,a)g(s)^2 s^{2\alpha} - \frac{\partial^2 G}{\partial z\,\partial w}(\rho,a)\rho g(s)s^{1+\alpha} + Q(s)s^{3\alpha}, \end{aligned} \tag{7}$$

for some function $Q(s)$, a Puiseux series with non-negative exponents.

In the case under study, the curve defined by G has a shape similar to the one depicted in Fig. 2, where

$$\frac{\partial G}{\partial w}(\rho, a) = 0. \tag{8}$$

This, together with the fact that $G(\rho, a) = 0$, shows that ρ is a root of the discriminant polynomial of G with respect to variable w, which is a polynomial in z (*cf.* [17]). In all the cases studied here, this polynomial has only one root in $]0, 1[$, a fact that allows to numerically get an approximation for the value of ρ. The minimum polynomial in $\mathbb{Q}[z]$ of ρ can be obtained by analysing the resultant

polynomials $G(z,w)$ and $\frac{\partial}{\partial w}G(z,w)$ with respect to w: $\mathrm{res}_w(G(z,w), \frac{\partial}{\partial w}G(z,w))$. We will denote this polynomial by $m(z)$. Using now the $\mathrm{res}_z(G(z,w), \frac{\partial}{\partial w}G(z,w))$ one can get a polynomial that has a as a root. One can then numerically compute all the real roots of that polynomial, and then check which one is an approximation for the value of a by means of a numerical study of the curve $G(z,w)$.

Using (8) in (7), and dividing it through by s^α, one gets

$$\begin{aligned} 0 = &-\frac{\partial G}{\partial z}(\rho,a)\rho s^{1-\alpha} + \frac{1}{2}\frac{\partial^2 G}{\partial z^2}(\rho,a)\rho^2 s^{2-\alpha} \\ &+ \frac{1}{2}\frac{\partial^2 G}{\partial w^2}(\rho,a)g(s)^2 s^\alpha + \\ &+ \frac{\partial^2 G}{\partial z\,\partial w}(\rho,a)\rho g(s)s + Q(s)s^{2\alpha}. \end{aligned} \tag{9}$$

Now, in all cases studied in this paper, one has

$$\frac{\partial G}{\partial z}(\rho,a) \neq 0, \text{ and } \frac{\partial^2 G}{\partial w^2}(\rho,a) \neq 0. \tag{10}$$

This was checked by computing

$$p_1(z) = \gcd_w(G(z,w), \frac{\partial}{\partial z}G(z,w)), \quad p_2(z) = \gcd_w(G(z,w), \frac{\partial^2}{\partial w^2}G(z,w)),$$

$\gcd(p_1(z), m(z))$ and $\gcd(p_2(z), m(z))$, obtaining a constant depending only on k, that is non-zero for all $k \neq 54$ in all cases dealt with in this paper. The case $k = 54$ was dealt with separately. Using the explicit value for ρ, the validity of (10) for this value of k was verified.

It now follows from (9), by noticing that the first and third summands have the smallest degrees in s, that they must have the same degree and cancel each other. Dividing, then, by s^α and letting $s \to 0$, one obtains

$$\alpha = \frac{1}{2}, \text{ and } b = g(0) = \sqrt{\frac{2\rho\,\frac{\partial G}{\partial z}(\rho,a)}{\frac{\partial^2 G}{\partial w^2}(\rho,a)}}.$$

In conclusion, for the case where $\lim_{z\to\rho} w(z) = a$, using (3), one has

$$[z^n]w(z) \sim \frac{b}{2\sqrt{\pi}}\rho^{-n}n^{-3/2}.$$

In the second case, the one where $\lim_{z\to\rho} w(z) = +\infty$, making $v = 1/w$ one concludes as above that $v = cs^\alpha - g(s)s^{\alpha+\beta}$, for some $0 < \alpha < 1$, $\beta > 0$, and for some Puiseux series $g(s)$, with non-negative exponents. Denoting by m the degree of G relative to w, the polynomial satisfied by v is then

$$H(z,v) = v^m G\left(z, \frac{1}{v}\right), \tag{11}$$

which is the reciprocal polynomial of $G(z,w)$ with respect to the variable w. Using the same procedure as above, one computes ρ, and checking that the corresponding

derivatives are non-zero, i.e.

$$\frac{\partial H}{\partial z}(\rho,0) \neq 0, \text{ and } \frac{\partial^2 H}{\partial w^2}(\rho,0) \neq 0,$$

one gets in the same way that

$$\alpha = \frac{1}{2}, \text{ and } c = \sqrt{\frac{2\rho \frac{\partial H}{\partial z}(\rho,0)}{\frac{\partial^2 H}{\partial w^2}(\rho,0)}}. \tag{12}$$

Since

$$\begin{aligned} w = \frac{1}{cs^\alpha - g(s)s^{\alpha+\beta}} &= \frac{1}{c}s^{-\alpha}\frac{1}{1-\frac{g(s)}{c}s^\beta} \\ &= \frac{1}{c}s^{-\alpha}\left(1+\frac{g(s)}{c}s^\beta + \frac{g(s)^2}{c^2}s^{2\beta}+\cdots\right), \end{aligned}$$

one sees, using (4), that

$$[z^n]w(z) \sim \frac{1}{c\sqrt{\pi}}\rho^{-n}n^{-1/2}. \tag{13}$$

5. Average Complexity of Regular Expressions in Strong Normal Form

Let $B_k(z)$ and $\bar{B}_k(z)$ be the generating functions for β_ε and $\beta_{\overline{\varepsilon}}$, as in (2), respectively. They satisfy the following equations

$$B_k(z) = 2zB_k(z)^2 + 2zB_k(z)\bar{B}_k(z) + 2z\bar{B}_k(z) \tag{14}$$

$$\bar{B}_k(z) = kz + 2zB_k(z)\bar{B}_k(z) + 2z\bar{B}_k(z)^2. \tag{15}$$

From (14) one gets

$$\bar{B}_k(z) = \frac{B_k(z)(1-2zB_k(z))}{2z(B_k(z)+1)},$$

and then substituting $\bar{B}_k(z)$ in (15) one obtains, after clearing up denominators,

$$4z^2B_k(z)^3 - (2kz^2+4z)B_k(z)^2 - (4kz^2-1)B_k(z) - 2kz^2 = 0,$$

i.e., $B_k(z)$ is an algebraic function that is a root of

$$4z^2w^3 - (2kz^2+4z)w^2 - (4kz^2-1)w - 2kz^2.$$

Using now (15) to get $B_k(z)$ as a function of $\bar{B}_k(z)$, and then substituting that into (14), one easily sees that $\bar{B}_k(z)$ is a root of

$$4zw^3 + 2kzw^2 - kw + k^2z.$$

Using the technique described in the previous section, one sees that $B_k(z)$ and $\bar{B}_k(z)$ have the same singularity, namely the only root (by using Sturm's Theorem), η_k, in the interval $]0,1[$ of the polynomial,

$$m_k(z) = z^3 + \frac{9z^2}{2k+27} - \frac{z}{8k+108} - \frac{1}{k(2k+27)}. \tag{16}$$

To show that $m_k(z)$ has no roots in the circle $|z| = \eta_k$, assume the contrary. Then, denoting by ζ one of the non-real roots, it would follow that $\eta_k^3 = \eta_k\zeta\bar{\zeta} = \frac{1}{2k^2+27k}$. Using this in $m_k(\eta_k) = 0$, one gets $\eta_k = \frac{1}{36}$, which does not hold for largely enough k (one can easily check that this never occurs!).

Also one gets that $\alpha = \frac{1}{2}$, and that the values of $a_{B_k} = B_k(\eta)$ and of $a_{\bar{B}_k} = \bar{B}_k(\eta)$ are roots of the polynomials $8z^3 - kz^2 + 2kz - k$, and $8z^3 + 2kz^2 - k^2$, respectively. With all this, and writing $S_k(z) = B_k(z) + \bar{B}_k(z)$ one then gets that

$$[z^n]S_k(z) \sim \frac{b_k}{2\sqrt{\pi}}\eta_k^{-n}n^{-3/2}, \tag{17}$$

where,

$$b_k = \frac{1}{2}\sqrt{\frac{4a_k^2(1-2a_k\eta_k) + k(1+4\eta_k-4a_k\eta_k)}{1-3a_k\eta_k}},$$

and

$$a_k = a_{B_k} + a_{\bar{B}_k}.$$

Using these results and the one mentioned in (5), the ratio of regular expressions in ssnf, $r_{k,n} = \frac{[z^n]S_k(n)}{[z^n]R_k(n)}$, can now be computed for any k and n. Using a technique that is beyond the scope of this paper, and that will be presented in a forthcoming article, it can be proved that

$$\eta_k \sim \frac{1}{\sqrt{8k}}, \qquad b_k \sim \sqrt{k}, \tag{18}$$

where $u_k \sim v_k$ means that the limit of the ratio of the two sequences $(u_k)_k$ and $(v_k)_k$ goes to 1, as k goes to infinity. We can, then, conclude that the ratio between the number of ssnf expressions and the number of general expressions, $r_{k,n}$, satisfies

$$\lim_{k\to\infty} r_{k,n} = 1, \text{ for all } n.$$

Since ρ_k is not a root of $m_k(z)$, we know that $\rho_k \neq \eta_k$. And, as $S_k(z)$ counts a subclass of regular expressions, it must follow that $\rho_k < \eta_k$, for all k. Thus,

$$\lim_{n\to\infty} r_{k,n} = 0, \text{ for all } k,$$

implying that for any given alphabet of k letters, the number of ssnf expressions is exponentially smaller than the number of all the all the expression, of the same size.

5.1. *Counting Letters*

To obtain the asymptotic average value of several measures for regular expressions of a given size, we consider bivariate generating functions parametrized by weights of the form c_o, with $o \in \{\emptyset, \varepsilon, \sigma, +, \cdot, \star, ?\}$, associated to each regular expression element. Considering the grammar in (2), let $B_k(u, z)$ and $\bar{B}_k(u, z)$ be the bivariate

generating functions associated to β_ε and $\beta_{\overline{\varepsilon}}$, respectively, and where u represents some given weight. Then

$$B_k(u,z) = (u^{c_\bullet} + u^{c_+})zB_k(u,z)^2 + 2u^{c_+}zB_k(u,z)\bar{B}_k(u,z) + (u^{c_?} + u^{c_\star})z\bar{B}_k(u,z),$$
$$\bar{B}_k(u,z) = ku^{c_\sigma}z + (u^{c_\bullet} + u^{c_+})z\bar{B}_k(u,z)^2 + 2u^{c_\bullet}zB_k(u,z)\bar{B}_k(u,z).$$

Note that A and B depend on the parameters $(c_\emptyset, c_\varepsilon, c_\sigma, c_+, c_\bullet, c_\star, c_?)$, but for sake of simplicity we choose to omit them. For computing the average number of letters those parameters are $(0,0,1,0,0,0,0)$, and analogously for each operator.

The generating function $L_k(z)$ for the number of letters is given by

$$L_k(z) = \left.\frac{\partial}{\partial u}\right|_{u=1} \left(B_k(u,z) + \bar{B}_k(u,z)\right).$$

Setting $B = B_k(1,z), \bar{B} = \bar{B}_k(1,z), B_1 = \left.\frac{\partial}{\partial u}\right|_{u=1} B_k(u,z), \bar{B}_1 = \left.\frac{\partial}{\partial u}\right|_{u=1} \bar{B}_k(u,z)$, one has:

$$\begin{aligned}
B &= 2B^2z + 2B\bar{B}z + 2\bar{B}z,\\
\bar{B} &= 2B\bar{B}z + 2\bar{B}^2z + kz,\\
B_1 &= 4BB_1z + 2B\bar{B}_1z + 2\bar{B}B_1z + 2\bar{B}_1z,\\
\bar{B}_1 &= 2B\bar{B}_1z + 2\bar{B}B_1z + 4\bar{B}\bar{B}_1z + kz,\\
L_k &= B_1 + \bar{B}_1.
\end{aligned}$$

Using Gröbner bases, as mentioned above, one gets the following polynomial for $w = L_k$:

$$\left(\left(8\,k^2 + 108\,k\right)z^3 + 36\,kz^2 - kz - 4\right)w^3 + \\ + \left(\left(k^3 + 12\,k^2\right)z^3 + 4\,k^2z^2 + kz\right)w - 2\,k^2z^3 - k^2z^2.$$

It turns out that, from this, one can deduce that the singularity for this algebraic function w has the same minimal polynomial as in (16), and so it is the same as for the number of regular expressions there considered. One then finds that, in this case, $\alpha = -\frac{1}{2}$, and that

$$[z^n]L_k(z) \sim \frac{1}{c_k\sqrt{\pi}}\eta_k^{-n}n^{-1/2}, \tag{19}$$

where, η_k is the same as in (17), and

$$c_k = \frac{2}{\sqrt{k}}\sqrt{\frac{8\sqrt{2}\,k^{5/2} + 112k^2 + 36\sqrt{2}\,k^{3/2} - 420k + 243\sqrt{2}\sqrt{k} - 81}{2\sqrt{2}\;k^{5/2} + 26k^2 + 11\sqrt{2}\,k^{3/2} - 73k + 36\sqrt{2}\sqrt{k} - 12}}.$$

Using the estimations (17) and (19), the density of letters in expressions of size n, $\frac{[z^n]L_k(n)}{n[z^n]S_k(n)}$, for any given k, is approximated by

$$\ell_k = \frac{2}{b_kc_k}.$$

Using that $c_k \sim \frac{4}{\sqrt{k}}$ and (18), one concludes that $\lim_{k\to\infty} \ell_k = \frac{1}{2}$.

5.2. *Size of ε-Follow Automata*

Considering the parameters $(2, 3, 3, -2, -1, 3, 1)$, as defined in Section 2, the generating function $F_k(z)$ for the size of the $\mathcal{A}_{\varepsilon\mathsf{f}}$ automaton is given by

$$F_k(z) = \frac{\partial}{\partial u}\bigg|_{u=1} \left(B_k(u,z) + \bar{B}_k(u,z)\right).$$

Using the same abbreviations as above, one has:

$$\begin{aligned}
B &= 2B^2z + 2B\bar{B}z + 2\bar{B}z\\
\bar{B} &= 2B\bar{B}z + 2\bar{B}^2z + kz\\
B_1 &= -3B^2z - 4B\bar{B}z + 4BB_1z + 2B\bar{B}_1z + 2\bar{B}B_1z + 4\bar{B}z + 2\bar{B}_1z\\
\bar{B}_1 &= -2B\bar{B}z + 2B\bar{B}_1z - 3\bar{B}^2z + 2\bar{B}B_1z + 4\bar{B}\bar{B}_1z + 3kz\\
F_k &= B_1 + \bar{B}_1.
\end{aligned}$$

$$\begin{aligned}
G(z,w) = {} & ((128\,k^2 + 1728\,k)z^5 + 576\,kz^4 - 16\,kz^3 - 64\,z^2)w^3 +\\
& + ((32\,k^3 + 432\,k^2)z^5 + (-16\,k^2 - 2160\,k)z^4 + (-4\,k^2 - 720\,k)z^3 +\\
& + 4\,kz^2 + 80\,z)w^2 + ((436\,k^3 + 5400\,k^2)z^5 + (-24\,k^3 + 1620\,k^2)z^4 +\\
& + (-110\,k^2 + 900\,k)z^3 + (3\,k^2 + 30\,k)z^2 + 6\,kz - 25)w + (81\,k^4 +\\
& + 350\,k^3 - 13500\,k^2)z^5 + (210\,k^3 - 5850\,k^2)z^4 + (-9\,k^3 - 645\,k^2)z^3 +\\
& + (-18\,k^2 + 250\,k)z^2 + 75\,kz
\end{aligned}$$

From this one can again deduce that the singularity for this algebraic function w still has exactly the same minimal polynomial as in(16).

Proceeding as above, one can verify that the singularity for $F_k(z)$ still has the same minimal polynomial as in (16), that $\alpha = -\frac{1}{2}$, and that

$$[z^n]F_k(z) \sim \frac{1}{d_k\sqrt{\pi}}\eta_k^{-n}n^{-1/2}, \tag{20}$$

where, η_k is still the same as in (17), and

$$d_k = \frac{4}{\sqrt{k}}\sqrt{\frac{16\sqrt{2}k^{7/2} + 192k^3 - 192\sqrt{2}k^{5/2} - 1288k^2 + 2174\sqrt{2}k^{3/2} - 2566k + 675\sqrt{2}\sqrt{k} - 135}{36\sqrt{2}k^{7/2} + 876k^3 - 364\sqrt{2}k^{5/2} - 30444k^2 + 31865\sqrt{2}k^{3/2} - 29089k + 6750\sqrt{2}\sqrt{k} - 1350}}.$$

The average ratio, $\frac{[z^n]F_k(n)}{n[z^n]S_k(n)}$, between the size of the $\mathcal{A}_{\varepsilon\mathsf{f}}$ and the size of the respective regular expression is approximated, using (17) and (20), one has

$$f_k = \frac{2}{b_k d_k}.$$

In a previous work [4], we were able to get an explicit expression, depending on k, for the asymptotic size of $\mathcal{A}_{\varepsilon\mathsf{f}}$, which allow us to conclude that its average complexity, in the case of general regular expressions, tends to $\frac{3}{4}$ of the size of the original expression, as k grows. In the present case, we had to use a different technique which allow us to obtain the above expression for d_k, from which it is

easy to conclude that $d_k \sim \frac{8}{3\sqrt{k}}$. From this it follows that the limit of f_k is $\frac{3}{4}$, as k goes to infinity. Thus the average size is only slightly more than half the worst-case size.

5.3. *Size of Glushkov Automata*

The number of states of the Glushkov automaton is the number of letters in the corresponding regular expression, whose average size was already computed in Subsection 5.1 and coincides with the one obtained for generic regular expressions. To estimate the average number of transitions of Glushkov automaton for regular expressions in strong star normal form, we consider the counting functions g and e given below, where g corresponds to the size of both *first* and *last*, and e corresponds to the size of *follow*.

$$\begin{aligned}
g(\varepsilon) = g(\emptyset) &= 0 & & \\
g(\sigma) &= 1 & e(\varepsilon) = e(\emptyset) = e(\sigma) &= 0;\\
g(\beta_1+\beta_2) &= g(\beta_1)+g(\beta_2) & e(\beta_1+\beta_2) &= e(\beta_1)+e(\beta_2)\\
g(\beta_\varepsilon\beta) &= g(\beta_\varepsilon)+g(\beta) & e(\beta_1\beta_2) &= e(\beta_1)+e(\beta_2)+g(\beta_1)\times g(\beta_2)\\
g(\beta_{\overline{\varepsilon}}\beta) &= g(\beta_{\overline{\varepsilon}}) & e(\beta_{\overline{\varepsilon}}^{\star}) &= e(\beta_{\overline{\varepsilon}})+g(\beta_{\overline{\varepsilon}})\times g(\beta_{\overline{\varepsilon}})\\
g(\beta_{\overline{\varepsilon}}^{\star}) &= g(\beta_{\overline{\varepsilon}}) & e(\beta_{\overline{\varepsilon}}^{?}) &= e(\beta_{\overline{\varepsilon}})\\
g(\beta_{\overline{\varepsilon}}^{?}) &= g(\beta_{\overline{\varepsilon}}) & &
\end{aligned}$$

To obtain the generating functions associated to these measures for expressions β, instead considering bivariate functions, we consider the cost generating functions $G_k(z) = \sum_\beta g(\beta) z^{|\beta|}$ and $E_k(z) = \sum_\beta e(\beta) z^{|\beta|}$. For example, one has

$$\begin{aligned}
G_k(z) &= \sum_\beta g(\beta) z^{|\beta|}\\
&= \sum_\sigma g(\sigma) z + \sum_{\beta_\varepsilon,\beta} (g(\beta_\varepsilon)+g(\beta)) z^{|\beta_\varepsilon\beta|}\\
&+ \sum_{\beta_{\overline{\varepsilon}},\beta} g(\beta_{\overline{\varepsilon}}) z^{|\beta_{\overline{\varepsilon}}\beta|} + \sum_{\beta_1,\beta_2} (g(\beta_1)+g(\beta_2)) z^{|\beta_1+\beta_2|}\\
&+ \sum_{\beta_{\overline{\varepsilon}}} g(\beta_{\overline{\varepsilon}}) z^{|\beta_{\overline{\varepsilon}}^{\star}|} + \sum_{\beta_{\overline{\varepsilon}}} g(\beta_{\overline{\varepsilon}}) z^{|\beta_{\overline{\varepsilon}}^{?}|}.
\end{aligned}$$

Considering also the corresponding generating functions restricted to expressions β_ε and $\beta_{\overline{\varepsilon}}$, one gets the following system of equations.

$$\begin{aligned}
G_k(z) &= kz + 3zG_k(z)S_k(z) + zG_k(z)S_{k,\varepsilon}(z) + 2zG_{k,\overline{\varepsilon}}(z)\\
G_{k,\overline{\varepsilon}}(z) &= kz + 2zG_{k,\overline{\varepsilon}}(z)S_k(z) + zG_k(z)S_{k,\overline{\varepsilon}}(z)\\
E_k(z) &= 4zE_k(z)S_k(z) + zG_k(z)^2 + 2zE_{k,\overline{\varepsilon}}(z) + zG_{k,\overline{\varepsilon}}(z)^2\\
E_{k,\overline{\varepsilon}}(z) &= 2zE_k(z)S_{k,\overline{\varepsilon}}(z) + 2zE_{k,\overline{\varepsilon}}(z)S_k(z) + 2zG_{k,\overline{\varepsilon}}(z)G_k(z) - zG_{k,\overline{\varepsilon}}(z)^2,
\end{aligned}$$

where $S_{k,\varepsilon} = B_k$ and $S_{k,\overline{\varepsilon}} = \bar{B}_k$.

This is a rather more complicated system of equations than the ones dealt with above, yielding that $w = G_k(z)$ satisfies

$$kp_k(z)w^3 - k^2z(1+2z)(5+2kz)w^2 + 2k^2zq_k(z)w - 4k^3z^2(1+2z) = 0,$$

while $w = E_k(z)$ satisfies

$$kzp_k(z)r_k(z)^2w^3 + k^2zp_k(z)s_k(z)w^2 + k^2t_k(z)w + 2k^4z^3u_k(z) = 0,$$

where $p_k(z) = 4k(2k+27)m_k(z)$ and $m_k(z)$ is the same as in (16), $q_k = 2k^2z^2 + 22kz^2 + 7kz + 2$, $r_k(z) = 18k^2z^3 + 250kz^3 + 85kz^2 - 2kz - 9$, and $s_k(z)$, $t_k(z)$, $u_k(z)$ have degrees 5, 10, 7, respectively. An analysis similar to the one conducted in the previous two sections allow us to conclude that

$$[z^n]G_k(z) \sim \frac{\mu_k}{2\sqrt{\pi}}\eta_k^{-n}n^{-\frac{3}{2}} \quad \text{and} \quad [z^z]E_k(z) \sim \frac{1}{\nu_k\sqrt{\pi}}\eta_k^{-n}n^{-\frac{1}{2}},$$

for some positive μ_k and ν_k, with $\nu_k \sim \frac{1}{\sqrt{k}}$. From this, and using (17), it follows that the average number of transitions per symbol of the original ssnf-regular expression, $\frac{[z^n](G_k(z)+E_k(z))}{n\,[z^n]S_k(z)} \sim 2$, with k, and for any given n.

6. Average Size of ssnf(α)

Given a general regular expression $\alpha \in \mathcal{R}_k$ we now estimate the average size of the corresponding expression in strong star normal form, i.e. the size of $\mathsf{ssnf}(\alpha)$. First we rewrite the grammar in (1) considering explicitly the rules for β_ε and $\beta_{\overline{\varepsilon}}$ in this general case:

$$\begin{aligned}
\alpha &:= \emptyset \mid \varepsilon \mid \beta,\\
\beta &:= \beta_\varepsilon \mid \beta_{\overline{\varepsilon}},\\
\beta_\varepsilon &:= \beta_\varepsilon\beta_\varepsilon \mid \beta_\varepsilon + \beta \mid \beta_{\overline{\varepsilon}} + \beta_\varepsilon \mid \beta^\star \mid \beta^?,\\
\beta_{\overline{\varepsilon}} &:= \sigma \mid \beta_{\overline{\varepsilon}}\beta \mid \beta_\varepsilon\beta_{\overline{\varepsilon}} \mid \beta_{\overline{\varepsilon}} + \beta_{\overline{\varepsilon}}.
\end{aligned} \tag{21}$$

From this we recall that the generating functions of β, β_ε and $\beta_{\overline{\varepsilon}}$, respectively R_k, $R_{k,\varepsilon}$, and $R_{k,\overline{\varepsilon}}$ satisfy the following equations [4]:

$$\begin{aligned}
R_k(z) &= kz + 2zR_k(z)^2 + 2zR_k(z)\\
R_{k,\varepsilon}(z) &= 2zR_{k,\varepsilon}(z)^2 + 2zR_{k,\varepsilon}(z)R_{k,\overline{\varepsilon}}(z) + 2zR_{k,\varepsilon}(z) + 2zR_{k,\overline{\varepsilon}}(z)\\
R_{k,\overline{\varepsilon}}(z) &= kz + 2zR_{k,\overline{\varepsilon}}(z)^2 + 2zR_{k,\varepsilon}(z)R_{k,\overline{\varepsilon}}(z)
\end{aligned}$$

Now using Definition 1, we obtain the cost functions c and m associated with the size of the expressions resulting from applying ssnf() and ss(), respectively.

$$
\begin{aligned}
c(\emptyset) = c(\varepsilon) &= c(\sigma) = 1\\
c(\beta_1+\beta_2) &= c(\beta_1)+c(\beta_2)+1\\
c(\beta_1\beta_2) &= c(\beta_1)+c(\beta_2)+1\\
c(\beta^\star) &= m(\beta)+1\\
c(\beta^?_\varepsilon) &= c(\beta_\varepsilon)\\
c(\beta^?_{\overline{\varepsilon}}) &= c(\beta_{\overline{\varepsilon}})+1
\end{aligned}
\qquad
\begin{aligned}
m(\emptyset) = m(\varepsilon) &= m(\sigma) = 1\\
m(\beta_1+\beta_2) &= m(\beta_1)+m(\beta_2)+1\\
m(\beta_\varepsilon\beta'_\varepsilon) &= m(\beta_\varepsilon)+m(\beta'_\varepsilon)+1\\
m(\beta_\varepsilon\beta_{\overline{\varepsilon}}) &= c(\beta_{\overline{\varepsilon}})+c(\beta_{\overline{\varepsilon}})+1\\
m(\beta_{\overline{\varepsilon}}\beta) &= c(\beta_{\overline{\varepsilon}})+c(\beta)+1\\
m(\beta^\star) &= m(\beta)\\
m(\beta^?) &= m(\beta)
\end{aligned}
$$

Let $C_k = \sum_\beta c(\beta)z^{|\beta|}$ and $M_k = \sum_\beta m(\beta)z^{|\beta|}$ be the cost generating functions associated with these measures and $X_{k,\varepsilon}$ and $X_{k,\overline{\varepsilon}}$ the respective functions when restricted to expressions β_ε and $\beta_{\overline{\varepsilon}}$, respectively and $X \in \{C, M\}$. We can omit the values for $\emptyset$ and ε as they do not contribute to the asymptotic estimates. As before, we have, for instance,

$$
\begin{aligned}
C_k(z) &= \sum_\beta c(\beta)z^{|\beta|}\\
&= \sum_\sigma c(\sigma)z + \sum_{\beta_1,\beta_2}(c(\beta_1)+c(\beta_2)+1)z^{|\beta_1+\beta_2|}\\
&+ \sum_{\beta_1,\beta_2}(c(\beta_1)+c(\beta_2)+1)z^{|\beta_1\beta_2|}\\
&+ \sum_\beta (m(\beta)+1)z^{|\beta^\star|} + \sum_{\beta_\varepsilon} c(\beta_\varepsilon)z^{|\beta^?_\varepsilon|} + \sum_{\beta_{\overline{\varepsilon}}}(c(\beta_{\overline{\varepsilon}})+1)z^{|\beta^?_{\overline{\varepsilon}}|}.
\end{aligned}
$$

From the above, the following system of equations is obtained:

$$
\begin{aligned}
C_k(z) &= kz + 4zC_k(z)R_k(z) + 2zR_k(z)^2 + zC_k(z) + zM_k(z) + zR_k(z) + zR_{k,\overline{\varepsilon}}(z),\\
M_k(z) &= kz + 2zM_k(z)R(z) + 2zR_k(z)^2 + 2zM_k(z) + 2zM_{k,\varepsilon}(z)R_{k,\varepsilon}(z) +\\
&\quad 2zC_{k,\overline{\varepsilon}}(z)R_{k,\varepsilon} + 2zC_k(z)R_{k,\overline{\varepsilon}}(z),\\
C_{k,\overline{\varepsilon}}(z) &= kz + 2zC_{k,\overline{\varepsilon}}(z)R_k(z) + 2zC_k(z)R_{k,\overline{\varepsilon}}(z) + 2zR_{k,\overline{\varepsilon}}(z)R_k(z),\\
C_{k,\varepsilon}(z) &= 2zC_k(z)R_{k,\varepsilon}(z) + 2zC_{k,\varepsilon}(z)R_k(z) + 2zR_k(z)R_{k,\varepsilon}(z) +\\
&\quad zC_k(z) + zM_k(z) + zR_k(z) + zR_{k,\overline{\varepsilon}}(z),\\
M_{k,\varepsilon}(z) &= 2zM_k(z) + 2zM_k(z)R_{k,\varepsilon}(z) + 2zM_{k,\varepsilon}(z)R_k(z) + 2zR_{k,\varepsilon}(z)R_k(z).
\end{aligned}
$$

Using Gröbner bases technique one obtains the following polynomial equation satisfied by $C = C_k(z)$:

$$2z(1+kz)pqC^2 + prC - kzs = 0, \tag{22}$$

where $p, q, r, s \in \mathbb{Z}[z]$ are given by

$$\begin{aligned}
p &= 1 - 4z + 4z^2 - 8kz^2,\\
q &= 1 - 4z + 3z^2 - 9kz^2 - 3kz^3 - 2k^2z^3,\\
r &= 1 + (-5+k)z + (8-13k)z^2 + (-4+12k-11k^2)z^3\\
&\quad + k(6+k-2k^2)z^4 + 4k^2z^5,\\
s &= 1 + (-5+k)z + (4-13k)z^2 + (16+10k-11k^2)z^3\\
&\quad + (-32+40k+k^2-2k^3)z^4 + 16(1-3k+k^2)z^5 - 16k^2z^6.
\end{aligned}$$

Solving (22) in order to C one sees, after some simplifications, that the generating function C is given by the root of this equation that has a definite value at $z = 0$, i.e.,

$$C = \frac{-r\sqrt{p} + |t|}{4z(1+kz)q\sqrt{p}},$$

where

$$\begin{aligned}
t &= 1 + (-7+k)z - 3(-6+5k)z^2 + (-20+42k-11k^2)z^3\\
&\quad + (8-46k+23k^2-2k^3)z^4 + 2k(6-27k+2k^2)z^5 - 8k^2(1+2k)z^6.
\end{aligned}$$

This can be further simplified yielding

$$C = \frac{2kzs}{\sqrt{p}\,(|t| + r\sqrt{p})}.$$

From this we see that the dominant singularity of $C_k(z)$ is either the positive root of $p(z)$, or the smallest positive root of $|t| + r\sqrt{p}$. The latter possibility can be discarded as follows. Let ρ_k be the positive root of p. Using Sturm's theorem, one ensures that each of the polynomials q, r and t has exactly one root in the interval $[0, \rho_k]$. Let ξ be the root of r in that interval. Using the first five terms of the appropriate Puiseux's expansion to approximate the root of r, one gets a vaule

$$\zeta_k = \frac{\left(\frac{1}{k}\right)^{2/3}}{\sqrt[3]{2}} - \frac{3}{2k} + \frac{19\left(\frac{1}{k}\right)^{4/3}}{6\ 2^{2/3}} - \frac{4}{3}2^{2/3}\left(\frac{1}{k}\right)^{5/3} + \frac{17}{12k^2},$$

for which $r(\zeta_k) > 0$, while $s(\zeta_k) < 0$ and $q(\zeta_k) < 0$, for a sufficiently large k. Therefore, for values in $[0, \xi_k]$, $|t| + r\sqrt{p}$ is always positive. In the interval $[\xi_k, \rho_k]$, since

$$t^2 - r^2 p = 8kz^2(1+kz)sq,$$

and r is negative, one sees that $|t| + r\sqrt{p}$ is also positive.

Applying, once again, the technique described at the end of the Section 4 one obtains

$$[z^n]C(z) \sim \frac{1}{c\sqrt{\pi}}\rho^{-n}n^{-1/2}. \tag{23}$$

where $\rho = \rho_k$ is, thus, the same as the one given in (5), and

$$c = c_k = 2\sqrt{\frac{(2k-1)(2u\sqrt{k}+v\sqrt{2})}{k^{3/2}(u'+2v'\sqrt{2k})}},$$

with

$$\begin{aligned} u &= 8k^5 - 12k^4 - 10k^3 + 15k^2 + 4k + 12, \\ v &= 8k^5 - 12k^4 + 22k^3 - 17k^2 - 20k - 4, \\ u' &= 32k^6 - 112k^5 + 112k^4 + 56k^3 + 66k^2 + 57k + 8, \\ v' &= 16k^5 - 32k^4 + 24k^3 - 64k^2 - 7k - 14. \end{aligned}$$

From (23), dividing it by the accumulated size of all regular expressions (using (5)) one can obtain the limit for the average reduction on the size of a regular expression rewritten as an ssnf expression

$$\gamma_k = \lim_{n\to\infty} \frac{[z^n]C_k(z)}{n\,[z^n]R_k(z)} = \frac{4\sqrt{\rho_k}}{c_k\sqrt[4]{2k}}. \tag{24}$$

One can easily check that this ratio goes to 1 when $k \to \infty$.

7. Experimental Results

We ran some experiments, using the FAdo package [20], to obtain average sizes of the measures studied above for small values of k and n. For the results to be statistically significant, regular expressions were uniformly random generated using a version of the grammar for $\mathcal{S}_k$ in reverse polish notation. For each size $n \in \{200, 500, 1000\}$, and alphabet size $k \in \{2, 10, 50\}$, samples of 10000 expressions were generated. This is sufficient to ensure a 95% confidence level within a 1% error margin. Table 1 presents the obtained average values of several measures for regular expressions in ssnf together with the asymptotic average values calculated, in Section 5, for the alphabetic size (ℓ_k) and the size of $\mathcal{A}_{\varepsilon\mathsf{f}}$ (f_k), respectively. The last column, labeled $wc_{\varepsilon\mathsf{f}}$, presents the worst case size of $\mathcal{A}_{\varepsilon\mathsf{f}}$ as given in Theorem 3, for expressions of size n. Table 2 shows results of the conversion of general expressions to ssnf and the asymptotic limit of the ratio calculated in the previous section (γ_k).

8. Conclusions

The average complexity results obtained for expressions in ssnf are slightly smaller than the ones obtained for general regular expressions, but the asymptotic limits as the alphabetic size goes to infinity are the same. Both the ratios, between the number of ssnf expressions and the number of general expressions, of a certain size, and between the size of ssnf expressions and the size of general expressions, tend to one[a].

[a]In the previous version of this paper, this limit was wrongly conjectured.

Table 1. Results for regular expressions in ssnf

k	$\lvert\alpha\rvert$	$\lvert\alpha\rvert_\Sigma$	$\lvert\delta_{\mathsf{pos}}\rvert$	$\frac{\lvert\delta_{\mathsf{pos}}\rvert}{\lvert\alpha\rvert}$	$\lvert\delta_{\varepsilon\mathsf{f}}\rvert$	$\lvert Q_{\varepsilon\mathsf{f}}\rvert$	$\lvert\varepsilon\mathsf{f}\rvert$	$\frac{\lvert\alpha\rvert_\Sigma}{\lvert\alpha\rvert}$	ℓ_k	$\frac{\lvert\varepsilon\mathsf{f}\rvert}{\lvert\alpha\rvert}$	f_k	$wc_{\varepsilon\mathsf{f}}$
2	200	83.86	348.69	1.74	112.20	52.86	165.06	0.42	**0.42**	0.83	**0.98**	1.479
	500	208.99	981.81	1.96	279.97	129.74	409.71	0.42		0.82		1.472
	1000	417.70	2048.25	2.05	559.04	257.85	816.89	0.42		0.82		1.469
	2000	834.90	4219.23	2.11	1117.90	513.83	1631.73	0.42		0.82		1.469
10	200	89.13	299.91	1.50	111.98	51.80	163.78	0.45	**0.44**	0.82	**0.90**	1.479
	500	222.09	823.34	1.65	279.11	126.91	406.02	0.44		0.81		1.472
	1000	443.77	1708.1	1.71	557.72	252.30	810.02	0.44		0.81		1.469
	2000	887.10	3490.5	1.75	1115.00	502.64	1617.64	0.44		0.81		1.469
50	200	93.63	254.89	1.27	108.53	51.29	159.82	0.47	**0.47**	0.80	**0.84**	1.479
	500	233.34	686.42	1.37	270.66	125.80	396.46	0.47		0.79		1.472
	1000	466.20	1412.69	1.41	540.84	249.94	790.78	0.47		0.79		1.469
	2000	931.97	2870.7	1.44	1081.10	498.14	1579.2	0.47		0.79		1.468

Table 2. Results of conversion to ssnf

k	$\lvert\alpha\rvert$	$\lvert\alpha\rvert_\Sigma$	$\lvert\mathsf{ssnf}(\alpha)\rvert$	$\frac{\lvert\mathsf{ssnf}(\alpha)\rvert}{\lvert\alpha\rvert}$	γ_k
2	200	67.1	175.20	0.88	**0.79**
	500	167.1	437.7	0.88	
	1000	333.8	875.4	0.88	
10	200	82.2	191.3	0.96	**0.93**
	500	204.7	478.2	0.96	
	1000	409.1	956.4	0.96	
50	200	91.36	197.68	0.99	**0.98**
	500	227.8	494.2	0.99	
	1000	455.0	988.3	0.99	
	2000	909.6	1976.8	0.99	

References

[1] V. M. Antimirov, Partial derivatives of regular expressions and finite automaton constructions., *Theoret. Comput. Sci.* **155**(2) (1996) 291–319.

[2] S. Broda, A. Machiavelo, N. Moreira and R. Reis, On the average state complexity of partial derivative automata: an analytic combinatorics approach, *Int. J. Found. Comput. Sci.* **22**(7) (2011) 1593–1606.

[3] S. Broda, A. Machiavelo, N. Moreira and R. Reis, On the average size of Glushkov and partial derivative automata, *Int. J. Found. Comput. Sci.* **23**(5) (2012) 969–984.

[4] S. Broda, A. Machiavelo, N. Moreira and R. Reis, A hitchhiker's guide to descriptional complexity through analytic combinatorics, *Theoret. Comput. Sci.* **528**(0) (2014) 85 – 100.
[5] S. Broda, A. Machiavelo, N. Moreira and R. Reis, Average size of automata constructions from regular expressions, *BEATCS* **116** (June 2015) 167–192.
[6] S. Broda, A. Machiavelo, N. Moreira and R. Reis, On the average complexity of strong star normal form, *Proc.19th DCFS 2017*, eds. C. Câmpeanu and G. Pighizzini *LNCS* **10316**, (Springer, 2017), pp. 77–88.
[7] A. Brüggemann-Klein, Regular expressions into finite automata, *Theoret. Comput. Sci.* **48** (1993) 197–213.
[8] J.-M. Champarnaud, F. Ouardi and D. Ziadi, Normalized expressions and finite automata, *Intern. Journ. of Alg. and Comp.* **17**(1) (2007) 141–154.
[9] J.-M. Champarnaud and D. Ziadi, Canonical derivatives, partial derivatives and finite automaton constructions, *Theoret. Comput. Sci.* **289** (2002) 137–163.
[10] P. Flajolet and R.Sedgewick, *Analytic Combinatorics* (CUP, 2008).
[11] V. M. Glushkov, The abstract theory of automata, *Russian Math. Surveys* **16**(5) (1961) 1–53.
[12] H. Gruber and S. Gulan, Simplifying regular expressions, *Proc. 4th LATA, 2010*, eds. A. Dediu, H. Fernau and C. Martín-Vide *LNCS* **6031**, (Springer, 2010), pp. 285–296.
[13] S. Gulan, On the relative descriptional complexity of regular expressions and finite automata, PhD thesis, Universität Trier (2011).
[14] S. Gulan and H. Fernau, Local elimination-strategies in automata for shorter regular expressions, *Proc. SOFSEM 2008, Vol. II*, eds. V. Geffert, J. Karhumäki, A. Bertoni, B. Preneel, P. Návrat and M. Bieliková (2008), pp. 46–57.
[15] E. Hille, *Analytic Function Theory* (Blaisdell Publishing Company, 1962).
[16] L. Ilie and S. Yu, Follow automata, *Inf. Comput.* **186**(1) (2003) 140–162.
[17] S. Lang, *Algebra*, Grad. Texts in Math.,, Vol. 211, 3rd edn. (Springer, New York, 2001).
[18] C. Nicaud, On the average size of Glushkov's automata, *Proc. 3rd LATA 2009*, eds. A. Dediu, A.-M. Ionescu and C. M. Vide *LNCS* **5457**, (Springer, 2009), pp. 626–637.
[19] G. Ott and N. H. Feinstein, Design of sequential machines from their regular expressions., *Journal of the ACM* **8**(4) (1961) 585–600.
[20] Project FAdo, tools for formal languages manipulation `http://fado.dcc.up.pt`, (Access date:1.1.2017).
[21] J. Sakarovitch, *Elements of Automata Theory* (CUP, 2009).
[22] K. Thompson, Regular expression search algorithm, *Communications of the ACM* **11**(6) (1968) 410–422.
[23] S. Yu, Regular languages, *Handbook of Formal Languages*, eds. G. Rozenberg and A. Salomaa, **1** (Springer, 1997).