An Introduction to Descriptional Complexity of Regular Languages through Analytic Combinatorics

Sabine Broda António Machiavelo Nelma Moreira Rogério Reis

Technical Report Series: DCC-2012-05
Version 1.0 July 2012

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
Rua do Campo Alegre, 1021/1055,
4169-007 PORTO,
PORTUGAL
Tel: 220 402 900 Fax: 220 402 950
http://www.dcc.fc.up.pt/Pubs/

An Introduction to Desriptional Complexity of Regular Languages through Analytic Combinatorics*

Sabine Broda, António Machiavelo, Nelma Moreira, Rogério Reis
sbb@dcc.fc.up.pt,ajmachia@fc.up.pt,{nam,rvr}@dcc.fc.up.pt
CMUP, Faculdade de Ciências da Universidade do Porto
Rua do Campo Alegre, 4169-007 Porto, Portugal

October 20, 2012

Abstract

Nowadays, an increasing attention is being given to the study of the descriptional complexity on the average case. Although the underlying theory for such a study seems intimidating, one can obtain interesting results in this area without too much effort. In this gentle introduction we take the reader on a journey through the basic analytical tools of that theory, giving some illustrative examples using regular expressions. We finalize with some new asymptotic average case results for several ε-NFA constructions, presented in a unified framework.

1 Introduction

Descriptional complexity studies the measures of complexity of languages and operations. Usually, the descriptional complexity of an object is the size of its shortest description which can be considered in the worst or average case. For each measure, it is important to know the size of the smallest representation for a given language, as well as how the size varies when several such representations are combined or transformed. These studies are motivated by the need to have good estimates of the amount of resources required to manipulate those representations. This is crucial in new applied areas where automata and other models of computation are used, for instance, for pattern matching in bioinformatics or network security, or for model checking or security certificates in formal verification systems. In general, having succinct objects will improve our control on software, which may become shorter, more efficient and easier to certify. Recently, the descriptional complexity of formal languages has been extensively researched; see [GKK$^+$02], [Hro02], [HK10b], [Yu05], [HK11], [Brz10], [HK10a], [YG11], and [Brz12].

Most studies of descriptional complexity as well as of computational complexity (or analysis of algorithms) consider worst-case analyses, for which well established methods are known [AHU74, Knu73a, Knu73b, Knu81]. However, a worst-case behavior can seldom occur and a worst-case upper bound can be of little use in practical applications. Think about the time a given trip can take if there is the chance of dead. Furthermore, the best

*This work was partially funded by the European Regional Development Fund through the programme COMPETE and by the Portuguese Government through the FCT under projects PEst-C/MAT/UI0144/2011 and CANTE-PTDC/EIA-CCO/101904/2008.

performing algorithms are not necessarily the ones with the best worst-case complexity. A classical example is the Quicksort algorithm. These facts motivate the study of complexity analysis in the average-case, where input data is assumed to follow a given probability of distribution.

Sedgewick and Flajolet [SF96] is a standard reference for the analysis of algorithms from the average-case point of view. One of the most important concepts used in this context is the notion of generating function. Given a set of objects C (*combinatorial class*) on which a non-negative integer function (size) $|\cdot|$ is defined, and such that for each $n \geq 0$, the number of objects of size n, c_n, is finite, the *generating function* $C(z)$ of C is the formal power series

$$C(z) = \sum_{c \in \mathsf{C}} z^{|c|} = \sum_{n=0}^{\infty} c_n z^n.$$

We denote by $[z^n]C(z)$ the coefficient of z^n, c_n.

The symbolic method (Flajolet and Sedgewick [FS08]) is a framework that allows the construction of a combinatorial class C in terms of simpler ones, $\mathsf{B}_1,\ldots,\mathsf{B}_n$, by means of specific operations, and such that the generating function $C(z)$ of C is a combination of the generating functions $B_i(z)$ of B_i, for $1 \leq i \leq n$. Such generating functions can be used, when considered as analytic functions over real or complex numbers, to obtain the exact or, more often, the asymptotic behavior of the coefficients. Multivariate generating functions can be used to simultaneously analyze different measures for a combinatorial class. The framework of analytic combinatorics [FS08] provides a powerful tool for asymptotic average-case analysis, by relating the enumeration of combinatorial objects to the algebraic and complex analytic properties of generating functions.

Among the formal languages, the regular languages are fundamental structures in computer science. Regular languages do not have a so called "perfect representation model" for which every desired manipulation, e.g. membership, minimality, equivalence, reverse, boolean operations, etc., is optimal. For instance, although regular expressions (REs) are a particularly powerful notation for regular language representation, even membership testing is an expensive operation. For deterministic finite automata (DFA), the membership testing is optimal, but DFAs simulate REs, or nondeterministic finite automata (NFA), with exponential cost. These non complexity preserving simulations by equivalent combinatorial models have been extensively studied, for the worst-case, in the literature [HK10a, HK10b, HK11]. Nicaud [Nic09] presented an average case study of the size of the Glushkov automata, proving that, on average, the number of transitions is linear in the size of the expression. This analysis was carried out using the framework of analytic combinatorics. Following the same approach, Broda *et al.* [BMMR11b, BMMR11a] proved that the size of the partial derivative automaton is on average half the size of the Glushkov automaton. Here we present the main mathematical tools necessary for such an analysis. We then illustrate the analytic combinatorial method, by deriving the asymptotic average-case complexities of several conversions between regular expressions and equivalent ε-automata.

In Section 2 we show, as a simple illustrative example, how one can use a generating function to obtain the exact number of words of a given size represented by a specific regular expression. In Section 3 we present the relevant notions and results from complex analysis required in what follows. The symbolic method is then summarized in Section 4. In Section 5 we use that method to compute the generating function corresponding to the regular expressions given by a particular grammar. Multivariate cost analysis is then addressed in Section 6, where it is applied to the example used throughout the article.

Finally, in the last section, the computation of the average case complexity of several ε-NFA constructions is presented.

2 Counting with Generating Functions

Suppose we want to know exactly how many words of size n are represented by the regular expression $(\mathtt{ab} + \mathtt{c} + \mathtt{d})^\star$.

It is easy to count this number of words for the first possible sizes. For $n = 0$ there is just one word: ε. For $n = 1$ there are two possibilities: `c` and `d`. Let us denote the number of words of size n by w_n. Thus we already know that $w_0 = 1$ and $w_1 = 2$. The value of w_n can be easily obtained from the values of w_{n-1} and w_{n-2}. Each word of size n either ends with a character `c` or `d` (Figure 1), in which case it results of the concatenation of a word of size $n-1$ with that character, or it ends with a character `b` in which case it must be the result of the concatenation of a word of size $n-2$ with the word `ab`. Thus each word of size $n-1$ originates two different words of size n and each word of size $n-2$ produces, concatenated with `ab`, just one word of size n that cannot be the concatenation of a word of size $n-1$ and a single character. Hence we have the following recurrence relation (for $n \geq 2$):

$$w_n = 2w_{n-1} + w_{n-2}. \tag{1}$$

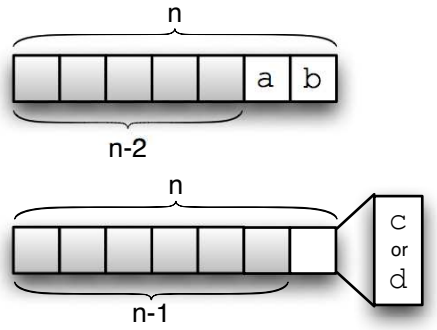

Figure 1: Generating the words of size n.

One way to obtain a closed formula for w_n from such a relation is through the use of the ring of formal power series. Multiplying (1) by z^n and "adding all" the resulting equalities, we get

$$\sum_{n\geq 2} w_n z^n = 2\sum_{n\geq 2} w_{n-1} z^n + \sum_{n\geq 2} w_{n-2} z^n. \tag{2}$$

Denoting by

$$W(z) = \sum_{n\geq 0} w_n z^n,$$

the generating function associated to $(w_n)_{n\in\mathbb{N}_0}$, we have

$$W(z) - (w_0 + w_1 z) = 2z(W(z) - w_0) + z^2 W(z),$$

and thus

$$(1 - 2z - z^2)W(z) = w_0 + w_1 z - 2w_0 z = 1,$$

since $w_0 = 1$ and $w_1 = 2$. Therefore

$$W(z) = \frac{1}{1 - 2z - z^2}. \tag{3}$$

Letting

$$\rho = -1 + \sqrt{2} \qquad \text{and} \qquad \bar{\rho} = -1 - \sqrt{2} \tag{4}$$

be the roots of the denominator of $W(z)$, we have

$$\begin{aligned} W(z) &= \frac{1}{2\sqrt{2}} \left(\frac{1}{\rho - z} - \frac{1}{\bar{\rho} - z} \right) \\ &= \frac{1}{2\sqrt{2}} \left(\frac{\frac{1}{\rho}}{1 - \frac{z}{\rho}} - \frac{\frac{1}{\bar{\rho}}}{1 - \frac{z}{\bar{\rho}}} \right) \\ &= \frac{1}{2\sqrt{2}} \left(\frac{1}{\rho} \sum_{n \geq 0} \left(\frac{z}{\rho} \right)^n - \frac{1}{\bar{\rho}} \sum_{n \geq 0} \left(\frac{z}{\bar{\rho}} \right)^n \right). \end{aligned}$$

Noticing that $\rho\bar{\rho} = -1$ we obtain

$$W(z) = \frac{1}{2\sqrt{2}} \sum_{n \geq 0} (-1)^{n+1} \left(\bar{\rho}^{n+1} - \rho^{n+1} \right) z^n,$$

and thus

$$w_n = \frac{1}{2\sqrt{2}} (-1)^{n+1} \left(\bar{\rho}^{n+1} - \rho^{n+1} \right),$$

which is, believe it or not, always a positive integer, as one could expect for the number of words of a given size. The previous formula can be rewritten as

$$w_n = \frac{(1 + \sqrt{2})^{(n+1)}}{2\sqrt{2}} \left(1 - (-1)^{n+1} (\sqrt{2} - 1)^{2n+2} \right),$$

which shows that

$$w_n = \frac{(1 + \sqrt{2})^{n+1}}{2\sqrt{2}} + o(1),$$

and gives the asymptotic behavior of w_n as

$$w_n \sim \frac{(1 + \sqrt{2})^{n+1}}{2\sqrt{2}} = \frac{1}{\sqrt{8}\,\rho} \rho^{-n}. \tag{5}$$

In this example it was easy to obtain an explicit relation like (1), and subsequently a closed formula as (2), but in general this is not the case. The technique explained in Section 4 permits to overcome the first problem, while complex analysis permits to estimate the asymptotic behavior without the need of a closed formula.

3 A Hike in Complex Analysis

Generating functions can be seen as complex analytic functions, and the study of their behavior around their dominant singularities gives us an asymptotic approximation to their

coefficients. In this section we will recall the basic notions and results that are crucial to the methods that follow.

In complex analysis one usually considers functions defined in *regions*, i.e. open connected subsets of $\mathbb{C}$. A complex function defined over a region Ω, $f : \Omega \to \mathbb{C}$, is said to be *analytic at* z_0 (with $z_0 \in \mathbb{C}$) if, for all z in some neighborhood of z_0 in Ω, f can be expressed by a convergent power series

$$f(z) = \sum_{n \geq 0} c_n (z - z_0)^n.$$

The function f is said to be *analytic* in the region Ω if it is analytic at every point of Ω. A differentiable function is also called *holomorphic*. A non-trivial result, due to Riemann, is that the two concepts, analytic and holomorphic, are equivalent.

An indispensable tool behind every result that follows is the Cauchy's coefficient formula, which is a direct corollary of his famous residue theorem. It states that if f is an analytic function in a region Ω containing 0, and if γ is a simple loop (with no self intersections) around 0 in Ω that is positively oriented, then

$$c_n = \frac{1}{2\pi i} \int_\gamma \frac{f(z)}{z^{n+1}} dz.$$

A *singularity* of a function f, defined in the region interior to a simple closed curve γ, is a point on γ such that f is not analytically continuable at that point, i.e. that cannot be extended by a continuous function to a neighborhood of that point. For example, the function $\sqrt{1-z}$ has singularity at 1.

Because in the context of analytic combinatorics all the series considered have non-negative integer coefficients, the result known as Pringsheim theorem [FS08, Theorem IV.6] is of particular importance, pinpointing a singularity of the generating function of a combinatorial class. This theorem states that, if f is representable at the origin by a series expansion with non-negative coefficients, and radius of convergence r, then the point $z = r$ is a singularity of f.

The following two theorems play a central role for obtaining an asymptotic approximation of the coefficients of certain generating functions. These are sufficient to all the problems addressed herein.

Theorem 1. *The coefficients of the function*

$$f(z) = (1 - z)^{-\alpha},$$

where $\alpha \in \mathbb{C} \setminus \mathbb{Z}_0^+$*, have the following asymptotic approximation:*

$$[z^n]\, f(z) = \frac{n^{\alpha-1}}{\Gamma(\alpha)} + o(n^{\alpha-1}).$$

And where Γ *is, as usual, the Euler's gamma function.*

This result can be shown in a more or less straightforward way using the binomial expansion and Stirling's formula, as pointed out in [FS08, pages 380 to 384]. Therein, this result is also shown using *Hankel contour* technique, which is likewise used to prove a theorem that ensures the transfer of an approximation of a function near a singularity into an asymptotic approximation of its coefficients.

For the next result we need the notion of Δ-*domain*, for which the definition follows and an illustration is presented in Figure 2.

Definition 2. *For $\xi \in \mathbb{C}$, $R > 1$ and $0 < \phi < \pi/2$, the* domain $\Delta(\xi, \phi, R)$ *is the open set*

$$\Delta(\xi, \phi, R) = \{z \in \mathbb{C} \mid |z| < R,\ z \neq \xi,\ \textit{and}\ |Arg(z - \xi)| > \phi\},$$

where $Arg(z)$ denotes the argument of $z \in \mathbb{C}$. A region is called a Δ-domain *at ξ if it is of the form $\Delta(\xi, \phi, R)$ for some ξ, ϕ, and R.*

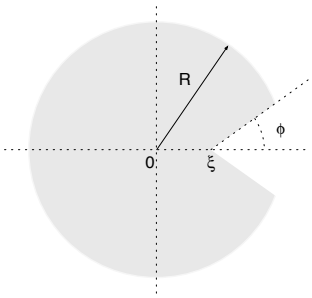

Figure 2: A Δ-domain.

Theorem 3. *Let $f(z)$ be a function defined in a Δ-domain at 1 satisfying*

$$f(z) = o\left((1 - z)^{-\alpha}\right).$$

Then,

$$[z^n]\, f(z) = o\left(n^{\alpha - 1}\right).$$

From these two last results, the next proposition easily follows, noting that $\Gamma(\frac{1}{2}) = \sqrt{\pi}$ and $\Gamma(-\frac{1}{2}) = -2\sqrt{\pi}$.

Proposition 4. *Let $f(z)$ be a function that is analytic in some Δ-domain at $\rho \in \mathbb{R}^+$. If at the intersection of a neighborhood of ρ and its Δ-domain,*

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

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

2. *$f(z) = \frac{a}{\sqrt{1 - z/\rho}} + o\left(\frac{1}{\sqrt{1 - z/\rho}}\right)$, with $a \in \mathbb{R}$, and $a \neq 0$, then*

$$[z^n] f(z) \sim \frac{a}{\sqrt{\pi}}\, \rho^{-n} n^{-1/2}.$$

4 The Symbolic Method

To describe a more general method to count families of combinatorial objects for which we do not have simple relations as in the example of Section 2, let us start with some basic notions.

Definition 5. *A combinatorial class $\mathcal{C}$ is a finite or enumerable set of objects on which a size function is defined, such that*

i. for every element $\gamma \in \mathcal{C}$, its size $|\gamma|$ is a non-negative integer;

ii. the number c_n of elements in $\mathcal{C}$ of size n is finite.

The sequence $c_0, c_1, c_2, \ldots$ is called the counting sequence of the class $\mathcal{C}$.

The generating function of the combinatorial class $\mathcal{C}$ is the formal series

$$\mathcal{G}(\mathcal{C}) = C(z) = \sum_{n \geq 0} c_n z^n = \sum_{\gamma \in \mathcal{C}} z^{|\gamma|}. \tag{6}$$

The symbolic method captures the fact that most of the basic set theoretic constructions over combinatorial classes translate, in a natural way, into operations on the corresponding generating functions. A combinatorial class with only one element, of size 1, is clearly represented by the generating function $U(z) = z$. Let $A(z) = \sum a_i z^i$ and $B(z) = \sum b_i z^i$ be generating functions of two combinatorial classes $\mathcal{A}$ and $\mathcal{B}$, respectively.

- If $\mathcal{A}$ and $\mathcal{B}$ are disjoint, then the class $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$ has $C(z) = A(z) + B(z)$ as generating function. This follows immediately from the fact that the number of objects in $\mathcal{C}$ of a given size is the sum of the number of the objects of the same size in $\mathcal{A}$ and $\mathcal{B}$.

- The cartesian product $\mathcal{C} = \mathcal{A} \times \mathcal{B}$, taking $|(a, b)| = |a| + |b|$, has generating function $C(z) = A(z)B(z)$, since
$$c_n = \sum_{i+j=n} a_i b_j.$$

- If $a_0 = 0$, the sequence class $\mathcal{C} = SEQ(\mathcal{A})$, which is defined as
$$SEQ(\mathcal{A}) = \{\varepsilon\} \cup \mathcal{A} \cup (\mathcal{A} \times \mathcal{A}) \cup (\mathcal{A} \times \mathcal{A} \times \mathcal{A}) \cup \cdots,$$
has as generating function
$$C(z) = \sum_{n \geq 0} A(z)^n = \frac{1}{1 - A(z)}.$$
This last equality follows from
$$(1 - A(z)) \left(\sum_{n \geq 0} A(z)^n \right) = \sum_{n \geq 0} A(z)^n - \sum_{n \geq 0} A(z)^{n+1} = 1,$$
noticing that these series are well defined because $A(z)$ has no independent term (alternatively, they converge in the ring of formal power series with the z-adic topology).

For other operations over combinatorial classes and their corresponding generating functions, see pages 26 to 31 of Flajolet & Sedgewick's book [FS08].

In the example given in Section 2, let $\mathcal{C}$ be the combinatorial class of all the words represented by $(\mathtt{ab} + \mathtt{c} + \mathtt{d})^\star$. The symbolic method immediately yields

$$\mathcal{G}(\mathcal{C}) = \frac{1}{1 - (U(z)U(z) + U(z) + U(z))} = \frac{1}{1 - 2z - z^2},$$

which is (3).

5 From Grammars to Generating Functions

Let us, now, consider the grammar for regular expressions proposed by Gruber and Gulan in [GG10], which has the major advantage of avoiding many redundant and unnatural expressions built with the symbols ε and $\emptyset$. Given an alphabet (set of *letters*) $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$ of size k, the set $\mathcal{A}_k$ of *regular expressions*, α, over Σ is defined by the following grammar,

$$\begin{aligned} &\alpha := \emptyset \mid \varepsilon \mid \beta \\ &\beta := \sigma_1 \mid \cdots \mid \sigma_k \mid (\beta + \beta) \mid (\beta \cdot \beta) \mid \beta^? \mid \beta^\star \end{aligned} \tag{7}$$

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

For the length 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. Equipped with the size function given by this definition of length of a regular expression, $\mathcal{A}_k$ is clearly a combinatorial class. The associated counting sequence is $a_0 = 0$ (there are no regular expressions with zero symbols), $a_1 = 2 + k$ ($\emptyset$, ε and the k symbols of Σ), $a_2 = 2k$ (the only two unary operators, ? and $\star$, applied to every symbol of Σ), $a_3 = 4k + 2k^2$ (each of the two unary operators applied to a regular expression of size 2, or two symbols of Σ operated by one of the two binary operators), etc...

Using the recursive definition of $\mathcal{A}_k$ in (7), we will now compute the associated generating function $A_k(z)$. Associated to the nonterminal symbol β of that grammar, we can consider the combinatorial class $\mathcal{B}_k$ with its corresponding generating function $B_k(z) = \sum b_n z^n$. In order to obtain the values of b_n, i.e. the number of expressions β of length n, note that β can be either a letter σ_i or of one of the forms $\beta+\beta$, $\beta\cdot\beta$, $\beta^?$ or $\beta^\star$. Since these are disjoint cases, they have to be counted separately using the principles of the symbolic method presented above, leading to the following equations

$$\begin{aligned} B_k(z) &= kU(z) + \mathcal{G}(\mathcal{B}_k \times \{+\} \times \mathcal{B}_k) + \\ &\qquad + \mathcal{G}(\mathcal{B}_k \times \{\cdot\} \times \mathcal{B}_k) + \mathcal{G}(\mathcal{B}_k \times \{^?\}) + \mathcal{G}(\mathcal{B}_k \times \{^\star\}) \; = \\ &= kU(z) + U(z)B_k(z)^2 + U(z)B_k(z)^2 + U(z)B_k(z) + U(z)B_k(z) \; = \\ &= kz + 2zB_k(z)^2 + 2zB_k(z). \end{aligned}$$

Solving this equation for $B_k(z)$ (as a complex function), one obtains two possible solutions

$$B_k(z) = \frac{-2z + 1 \pm \sqrt{(4-8k)z^2 - 4z + 1}}{4z}.$$

But, since $B_k(0) = b_0 = a_0 = 0$, one must have $\lim_{z\to 0} B_k(z) = 0$, which is satisfied only by

$$B_k(z) = \frac{-2z + 1 - \sqrt{(4-8k)z^2 - 4z + 1}}{4z}.$$

Concerning $A_k(z)$, there are two expressions, $\emptyset$ and ϵ, of size 1, plus the whole class $\mathcal{B}_k$ with the generating function $B_k(z)$. Thus, we have

$$A_k(z) = U(z) + U(z) + B_k(z) = 2z + B_k(z),$$

and consequently

$$A_k(z) = \frac{8z^2 - 2z + 1 - \sqrt{(4-8k)z^2 - 4z + 1}}{4z}. \tag{8}$$

Let

$$\Delta_k(z) = (4-8k)z^2 - 4z + 1 \tag{9}$$

which has the following two zeros

$$\rho_k = \frac{1}{2(\sqrt{2k}+1)} \qquad \overline{\rho_k} = \frac{-1}{2(\sqrt{2k}-1)}.$$

Consider now

$$F_k(z) = 4zA_k(z) - 8z^2 + 2z = 1 - \sqrt{\Delta_k(z)}, \tag{10}$$

which has ρ_k as the only singularity. To deduce the asymptotic behavior of $F_k(z)$ from Proposition 4, observe that, if f is a complex function defined in a neighborhood of ρ such that $\lim_{z\to\rho} f(z) = a$, one has, for all $r \in \mathbb{R}$,

$$f(z)\left(1 - \frac{z}{\rho}\right)^r = a\left(1 - \frac{z}{\rho}\right)^r + o\left(\left(1 - \frac{z}{\rho}\right)^r\right), \tag{11}$$

Now,

$$\begin{aligned} \Delta_k(z) &= (4-8k)(z-\overline{\rho_k})(z-\rho_k) \\ &= (8k-4)(z-\overline{\rho_k})\rho_k(1-z/\rho_k) \end{aligned} \tag{12}$$

and

$$(8k-4)(\rho_k - \overline{\rho_k})\rho_k = 4\sqrt{2k}\rho_k. \tag{13}$$

Thus, by the previous observation, one has

$$\sqrt{\Delta_k(z)} = 2\sqrt[4]{2k}\sqrt{\rho_k}\sqrt{1-z/\rho_k} + o\left(\sqrt{1-z/\rho_k}\right),$$

and consequently,

$$F_k(z) = 1 - 2\sqrt[4]{2k}\sqrt{\rho_k}\sqrt{1-z/\rho_k} + o\left(\sqrt{1-z/\rho_k}\right). \tag{14}$$

By Proposition 4, one obtains

$$[z^n]F_k(z) \sim \frac{\sqrt[4]{2k}\sqrt{\rho_k}}{\sqrt{\pi}}\rho_k^{-n} n^{-3/2}. \tag{15}$$

From (10) one concludes that, for $n \geq 2$,

$$[z^n]A_k(z) \sim \frac{\sqrt[4]{2k}\sqrt{\rho_k}}{4\sqrt{\pi}}\rho_k^{-(n+1)}(n+1)^{-3/2} \tag{16}$$

Table 1 contains values of the ratio between the computed approximation and the actual coefficients of the power series of $A_k(z)$ for different values of k and n.

Thus we have obtained a very good approximation for the number of regular expressions with a given number of symbols, even for very small alphabets.

k \ n	10	20	100	500
2	0.95	0.97	0.99	1.00
10	0.95	0.97	0.99	1.00
50	1.06	0.98	0.99	1.00
200	1.44	1.11	0.99	1.00

Table 1: Accuracy of the approximation

6 Multivariate Cost Functions and Average-Case Analysis

For a given combinatorial class $\mathcal{C}$, one may be interested not only in counting the number of objects of a given size, but also considering some other features, or costs, of these objects. For example, one can be interested in the average number of occurrences of a given character in the words of fixed size represented by a certain regular expression. This can be achieved by considering multi-indexed sequences, that can be dealt with using multivariate generating functions. Considering t cost functions, $p_1, p_2, \ldots, p_t : \mathcal{C} \to \mathbb{C}$, let $c_{k_1,\ldots,k_t,n}$ be the number of objects c of size n with $p_1(c) = k_1, \ldots, p_t(c) = k_t$. This gives rise to the following multivariate cost generating function

$$C(u_1, \ldots, u_t, z) = \sum_{n,k_1,\ldots,k_t \geq 0} c_{k_1,\ldots,k_t,n}\, u_1^{k_1} \cdots u_t^{k_t} z^n.$$

The functions $(p_i)_i$ give the respective weights of the t features under consideration.

Note that since $\mathcal{C}$ is a combinatorial class, the number of objects with a given size is finite, and therefore, for a fixed n, there are only a finite number of $c_{k_1,\ldots,k_t,n}$ which are different from 0. Also,

$$\left.\frac{\partial C(u_1, \ldots, u_t, z)}{\partial u_i}\right|_{u_i=1} = \sum_{\substack{n,k_j \geq 0 \\ j \neq i}} \left(\sum_{k_i \geq 0} k_i c_{k_1,\ldots,k_t,n} \right) u_1^{k_1} \cdots u_{i-1}^{k_{i-1}} u_{i+1}^{k_{i+1}} \cdots u_t^{k_t} z^n,$$

where

$$\sum_{k_i \geq 0} k_i c_{k_1,\ldots,k_t,n}$$

accounts for the cumulative presence of cost p_i in the objects of size n.

For the language given as example in Section 2, suppose we want to find out the average number of occurrences of the symbol $\mathtt{a}$ in the words of a certain size. In order to do that let us consider the multivariate generating function that counts the number of occurrences of the symbol $\mathtt{a}$ in the words represented by $(\mathtt{ab} + \mathtt{c} + \mathtt{d})^\star$. This can be calculated using the symbolic method explained in Section 4 for one variable and that can be easily seen to still hold for multivariate functions. Since

$$\mathcal{G}(\mathtt{a}) = uz \quad \mathcal{G}(\mathtt{b}) = z \quad \mathcal{G}(\mathtt{c}) = z \quad \mathcal{G}(\mathtt{d}) = z,$$

one has

$$\mathcal{G}(\mathtt{ab} + \mathtt{c} + \mathtt{d}) = 2z + uz^2,$$

and therefore

$$\mathcal{G}((\mathtt{ab} + \mathtt{c} + \mathtt{d})^\star) = \frac{1}{1 - (2z + uz^2)}. \tag{17}$$

Now

$$\begin{aligned}\frac{1}{1-(2z+uz^2)} &= \sum_{n\geq 0}\left(2z+uz^2\right)^n = \sum_{n\geq 0}(2+uz)^n z^n \\ &= \sum_{n\geq 0}\left(\sum_{k=0}^{n}\binom{n}{k}2^{n-k}u^k z^k\right)z^n \\ &= \sum_{n\geq 0}\left(\sum_{k=0}^{n}\binom{n}{k}2^{n-k}u^k z^{n+k}\right) \\ &= \sum_{m\geq 0}\left(\sum_{n+k=m}\binom{n}{k}2^{n-k}u^k\right)z^m.\end{aligned}$$

It follows that

$$\mathcal{G}((\mathtt{ab}+\mathtt{c}+\mathtt{d})^\star) = \sum_{m\geq 0}\left(\sum_{k=0}^{m}\binom{m-k}{k}2^{m-2k}u^k\right)z^m,$$

which means that the number of words of length m with k occurrences of the symbol $\mathtt{a}$ is $\binom{m-k}{k}2^{m-2k}$.

Note that, according to what was explained above, the total number of occurrences of $\mathtt{a}$ in words of length m is exactly given by the derivative, at $u=1$, of

$$\sum_{k=0}^{m}\binom{m-k}{k}2^{m-2k}u^k,$$

which is equal to

$$\sum_{k=0}^{m}k\binom{m-k}{k}2^{m-2k}.$$

This, although a closed formula, gives no hint on how to obtain an asymptotic approximation[1], for, for example, the estimation of the average occurrences of $\mathtt{a}$ in the words of the language. It is at this point that complex analysis shows its usefulness.

The partial derivative of (17) with respect to u evaluated at $u=1$ is

$$A(z) = \frac{z^2}{(1-2z-z^2)^2}.$$

Using the observation made in (11), it follows that

$$\begin{aligned}A(z) &= \frac{1}{(\bar{\rho}-\rho)^2}\left(1-\frac{z}{\rho}\right)^{-2} + o\left(\left(1-\frac{z}{\rho}\right)^{-2}\right) \\ &= \frac{1}{8}\left(1-\frac{z}{\rho}\right)^{-2} + o\left(\left(1-\frac{z}{\rho}\right)^{-2}\right),\end{aligned}$$

for ρ and $\bar{\rho}$ as in (4). The obvious adaptations of Theorems 1 and 3 yield

$$a_n = [z^n]A(z) \sim \frac{n}{8\,\Gamma(2)}\rho^{-n} = \frac{n}{8}\rho^{-n}. \tag{18}$$

[1] As a matter of fact, in this case, with some work, it is possible to obtain a simpler, general expression for the coefficients of the series that makes evident their asymptotic behavior. See A.

The percentage of the symbol a in the words of size n is obviously given by

$$\frac{a_n}{n\,w_n},$$

i.e. the total number of occurrences of the symbol a over the total number of letters in the words of size n. From (18) and (5) we can conclude that

$$\lim_{n\to\infty} \frac{a_n}{n\,w_n} = \frac{1}{4+\sqrt{8}}\,,$$

which is the asymptotic limit of the percentage of symbols a in the words of the considered language. The fact that this limit is not rational was certainly not easily foreseeable at the beginning of this problem!

7 An Example: Comparing Different ε-NFA Constructions

Regular languages can also be defined by finite automata. The conversion from a regular expression to an equivalent finite automaton is at the basis of most techniques used in applications dealing with text. This makes comparing the complexity of such conversion algorithms an issue of great significance. Furthermore, for practical purposes, the average case complexity of such a construction is much more relevant than its worst-case complexity, which is often due to some particular and rarely occurring cases.

In this section we compare three different conversion algorithms in terms of the size of the resulting automata. These exhibit significantly different performances, as we will see, using the cost generating function approach explained in the previous section.

A *non-deterministic finite automaton* (NFA) is a tuple $\mathcal{N} = (Q, \Sigma, \delta, q_0, F)$ 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 $|\mathcal{N}| = |Q| + |\delta|$, the number of states $|\mathcal{N}|_Q = |Q|$, and the number of transitions $|\mathcal{N}|_\delta = |\delta|$. An NFA that has transitions labelled with ε is an ε-NFA. Standard definitions and results on this subject, such as the language accepted by an NFA, *etc.*, can be found in any good textbook on formal language theory, but are not relevant to our purpose.

7.1 Three Conversion Algorithms

We consider here three constructions, introduced respectively by Thompson in 1968 [Tho68], by Sippu and Soisalon-Soininen in 1990 [SSS90], and by Ilie and Yu in 2003 [IY03], that transform a regular expression α into an equivalent ε-NFA. Denoting the result by $\mathcal{N}_\alpha$, all three algorithms associate to the (atomic) regular expressions $\emptyset$, ε and σ the same ε-NFAs, as given in Figure 3.

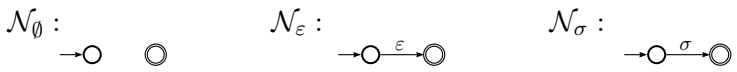

Figure 3: ε-NFAs for atomic expressions

Thus, for all three constructions we have

$$\begin{array}{lll} |\mathcal{N}_\emptyset| &= |\mathcal{N}_\emptyset|_Q + |\mathcal{N}_\emptyset|_\delta &= 2+0 \quad = 2 \\ |\mathcal{N}_\varepsilon| &= |\mathcal{N}_\varepsilon|_Q + |\mathcal{N}_\varepsilon|_\delta &= 2+1 \quad = 3 \\ |\mathcal{N}_\sigma| &= |\mathcal{N}_\sigma|_Q + |\mathcal{N}_\sigma|_\delta &= 2+1 \quad = 3 \end{array}$$

The ε-NFA's for compound regular expressions are then constructed inductively from the automata corresponding to their subexpressions.

In the Thompson's construction, the automaton $\mathcal{N}_{\beta_1+\beta_2}$ (in Figure 4) is built from $\mathcal{N}_{\beta_1}$ and $\mathcal{N}_{\beta_2}$ introducing a new initial state with ε-transitions to the initial states of both $\mathcal{N}_{\beta_1}$ and $\mathcal{N}_{\beta_2}$, as well as a new final state and ε-transitions from the final states of $\mathcal{N}_{\beta_1}$ and $\mathcal{N}_{\beta_2}$. It follows that this construction introduces exactly 2 states and 4 transitions for each + operator. Thus, we have

$$\begin{aligned} |\mathcal{N}_{\beta_1+\beta_2}|_Q &= |\mathcal{N}_{\beta_1}|_Q + |\mathcal{N}_{\beta_2}|_Q + 2 \\ |\mathcal{N}_{\beta_1+\beta_2}|_\delta &= |\mathcal{N}_{\beta_1}|_\delta + |\mathcal{N}_{\beta_2}|_\delta + 4, \end{aligned} \tag{19}$$

and consequently,

$$|\mathcal{N}_{\beta_1+\beta_2}| = |\mathcal{N}_{\beta_1}| + |\mathcal{N}_{\beta_2}| + 6. \tag{20}$$

The remaining construction cases are presented in Figure 4. Although the Thompson construction is usually presented without the result for the option operator ?, we included it here with a construction for $\mathcal{N}_{\beta?}$ that saves 2 states and 2 transitions relative to $\mathcal{N}_{\beta+\varepsilon}$.

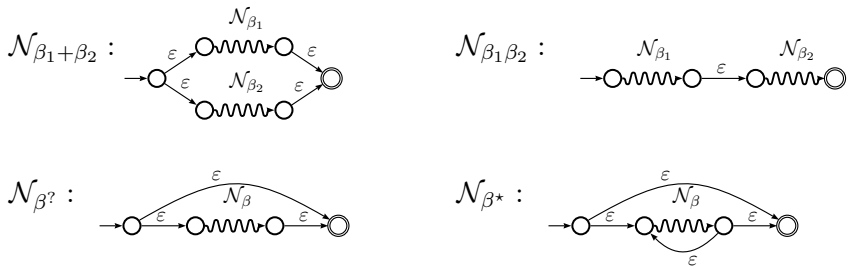

Figure 4: Thompson construction

Observe that the ε-NFAs resulting from the Thompson construction have the following properties:

- the number of states is even;
- the initial state is non-returning;
- there is only one final state, which is non-exiting;
- each state has at most two ingoing transitions, and at most two outgoing transitions.

In Figure 5 and Figure 6 we present the remaining cases for the other two conversion algorithms. Some of the above properties do not hold for these two other constructions.

For any of these constructions, and in each one of the corresponding construction cases, the size of the resulting automaton equals the sum of the sizes of its constituents plus some constant, as in (19) and (20).

7.2 Average Size of the ε-NFAs

To obtain the asymptotic average size of the automata resulting of each of these three constructions, we will parametrize them in a single unifying framework. This approach has the advantage of being easily reusable for a similar study of other regular-expression-to-NFA transformations. In order to do so, let us consider the different parameters for the

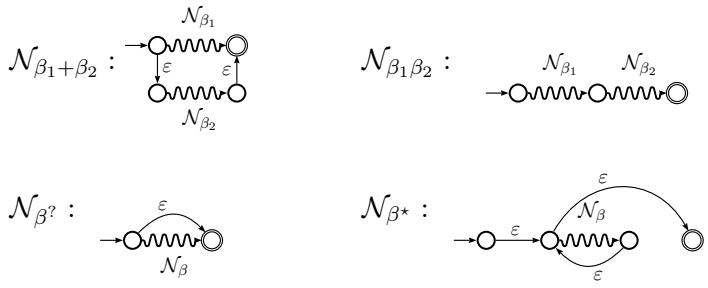

Figure 5: Sippu & Soisalon-Soininen (SSS) construction

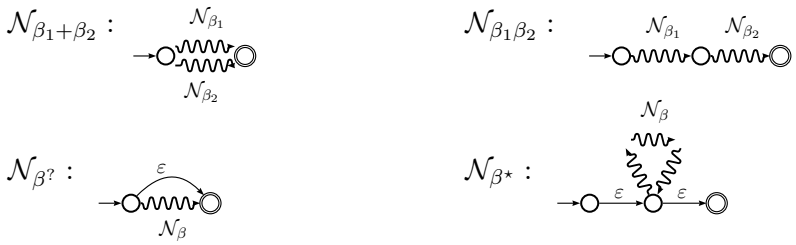

Figure 6: ε-follow construction by Ilie and Yu

expressions, analogous to (19) and (20), pertaining to the considered constructions. For each transformation we need to consider the following parameters $(c_\emptyset, c_\varepsilon, c_\sigma, c_+, c_\bullet, c_?, c_\star)$, for the obvious corresponding operations (where $c_\bullet$ corresponds to the concatenation). For the three conversion algorithms described in the previous subsection, and for the three considered measures, the values of these parameters are given in Table 2.

ε-NFAs	States	Transitions	Combined Size
Thompson	(2,2,2,2,0,2,2)	(0,1,1,4,1,3,4)	(2,3,3,6,1,5,6)
SSS	(2,2,2,0,-1,0,2)	(0,1,1,2,0,1,3)	(2,3,3,2,-1,1,5)
ε-follow	(2,2,2,-2,-1,0,1)	(0,1,1,0,0,1,2)	(2,3,3,-2,-1,1,3)

Table 2: Characteristic constants for the 3 constructions

For normalized regular expressions, given by the grammar (7), and for a particular transformation from regular expressions to ε-NFAs, the cost generating function of the resulting automata, according to a given measure, can be obtained, using the symbolic method, as follows

$$\begin{aligned}
\mathcal{G}(\sigma) &= u^{c_\sigma} z \\
\mathcal{G}(\beta + \beta) &= \mathcal{G}(\beta)\, u^{c_+} z\, \mathcal{G}(\beta) \\
\mathcal{G}(\beta \cdot \beta) &= \mathcal{G}(\beta)\, u^{c_\bullet} z\, \mathcal{G}(\beta) \\
\mathcal{G}(\beta^?) &= \mathcal{G}(\beta)\, u^{c_?} z \\
\mathcal{G}(\beta^\star) &= \mathcal{G}(\beta)\, u^{c_\star} z,
\end{aligned}$$

and thus

$$\mathcal{G}(\beta) = kzu^{c_\sigma} + (u^{c_+} + u^{c_\bullet}) z \mathcal{G}(\beta)^2 + (u^{c_?} + u^{c_\star}) z \mathcal{G}(\beta).$$

Solving this equation with respect to $\mathcal{G}(\beta)$, and using the binomial expansion to determine

which root has positive coefficients, one sees that

$$\mathcal{G}(\beta) = \frac{1-(u^{c_?}+u^{c_\star})z - \sqrt{(1-(u^{c_?}+u^{c_\star})z)^2 - 4ku^{c_\sigma}(u^{c_+}+u^{c_\bullet})z^2}}{2(u^{c_+}+u^{c_\bullet})z},$$

which henceforth will be denote by $C_k(u,z)$.

Deriving in order to u and taking $u=1$, the cumulative generating function obtained is

$$C_k(z) = \frac{a_k(z)\sqrt{\Delta_k(z)}+b_k(z)}{8z\sqrt{\Delta_k(z)}},$$

where $\Delta_k(z)$ is given by (9), and

$$\begin{aligned} a_k(z) &= 2(c_+ + c_\bullet - c_\star - c_?)z - (c_+ + c_\bullet) \\ b_k(z) &= 4(2kc_\sigma - c_\star - c_? + (1-k)(c_+ + c_\bullet))z^2 + \\ &\quad +2(c_\star + c_? - 2c_+ - 2c_\bullet)z + c_+ + c_\bullet. \end{aligned}$$

Hence, the cumulative generating function corresponding to the α symbol of the grammar in (7) is given by $D_k(u,z) = zu^{c_\emptyset} + zu^{c_\varepsilon} + C_k(u,z)$. Therefore

$$D_k(z) = \left.\frac{\partial D_k(u,z)}{\partial u}\right|_{u=1} = (c_\varepsilon + c_\emptyset)z + C_k(z) = \frac{d_k(z)\sqrt{\Delta_k(z)}+b_k(z)}{8z\sqrt{\Delta_k(z)}},$$

where

$$\begin{aligned} d_k(z) &= 8(c_\varepsilon + c_\emptyset)z^2 + d_k(z) \\ &= 8(c_\varepsilon + c_\emptyset)z^2 + 2(c_+ + c_\bullet - c_\star - c_?)z - (c_+ + c_\bullet). \end{aligned}$$

As was done in Section 5, and in order to apply Proposition 4, we consider the function

$$G_k(z) = 8zD_k(z) - d_k(z) = \frac{b_k(z)}{\sqrt{\Delta_k(z)}}. \tag{21}$$

From (12), one gets

$$G_k(z) = \frac{b_k(z)}{\sqrt{(8k-4)(z-\overline{\rho_k})\rho_k}\,\sqrt{1-z/\rho_k}}.$$

Since, by (13),

$$\lim_{z\to\rho_k} \frac{b_k(z)}{\sqrt{(8k-4)(z-\overline{\rho_k})\rho_k}} = \frac{b_k(\rho_k)}{2\sqrt[4]{2k}\sqrt{\rho_k}},$$

one has by (11)

$$G_k(z) = \frac{b_k(\rho_k)}{2\sqrt[4]{2k}\sqrt{\rho_k}\,\sqrt{1-z/\rho_k}} + o\left(\frac{1}{\sqrt{1-z/\rho_k}}\right).$$

From Proposition 4, it follows that

$$[z^n]G_k(z) \sim \frac{b_k(\rho_k)}{2\sqrt[4]{2k}\sqrt{\rho_k}\sqrt{\pi}}\,\rho_k^{-n}n^{-\frac{1}{2}}.$$

Finally, one concludes from (21) that, for $n \geq 1$,

$$[z^n]D_k(z) \sim \frac{b_k(\rho_k)}{16\sqrt[4]{2k}\sqrt{\rho_k}\sqrt{\pi}}\ \rho_k^{-(n+1)}(n+1)^{-\frac{1}{2}}. \tag{22}$$

The following expression gives, for $n \geq 2$, the asymptotic estimate for the average size for any automaton construction for regular expressions of size n.

$$\begin{aligned}\frac{[z^n]D_k(z)}{[z^n]A_k(z)} &\sim \frac{b_k(\rho_k)}{4\sqrt{2k}\ \rho_k}\ (n+1) = \\ &= \frac{\sqrt{2}(2kc_\sigma + \sqrt{2k}(c_\star + c_?) + k(c_+ + c_\bullet))}{4(\sqrt{2}\ k + \sqrt{k})}\ (n+1)\end{aligned}$$

Thus, the ratio between the size of the ε-NFA and the size of the original regular expression is given by

$$\lim_{n\to\infty} \frac{[z^n]D_k(z)}{n[z^n]A_k(z)} = \frac{\sqrt{2}(k(2c_\sigma + c_+ + c_\bullet) + \sqrt{2k}(c_\star + c_?))}{4(\sqrt{2}\ k + \sqrt{k})}\ . \tag{23}$$

Now, note that for all the constructions here considered, the worst-case complexity is reached for expressions with only one letter and $n-1$ stars. For such expression (that has size n), the size of the corresponding Thompson, SSS and ε-follow automaton is, respectively, $6n-3$, $5n-2$ and $3n$. In Table 3, we illustrate the discrepancy between the average and the worst-case, by presenting the values of the equation (23) for different values of k, the limit as k goes to infinity, and the worst-case, that does not depend on k.

k	2	10	50	100	∞	worst-case
Thompson	4	3.7	3.5	3.4	3.25	6
SSS	2.2	2.0	1.9	1.8	1.75	5
ε-follow	1.2	1.0	0.9	0.8	0.75	3

Table 3: Average vs. worst-case

7.3 Counting ε-Transitions

We can also use the above cost generating function in order to estimate the average number of ε-transitions introduced by the three conversion algorithms. This is of particular interest for practical applications, since ε-transitions create delay in pattern matching. The corresponding tuples for this analysis are given in Table 4.

Let $D_k^\varepsilon(z)$ and $D_k^\delta(z)$ be, respectively, the generating functions for the number of ε-transitions and for all transitions in an ε-NFA. From (22) or (23) one obtains the following expression for the asymptotic density of ε-transitions

$$\lim_{n\to\infty} \frac{[z^n]D_k^\varepsilon(z)}{[z^n]D_k^\delta(z)} = \frac{k(2c_\sigma^\varepsilon + c_+^\varepsilon + c_\bullet^\varepsilon) + \sqrt{2k}(c_\star^\varepsilon + c_?^\varepsilon)}{k(2c_\sigma^\delta + c_+^\delta + c_\bullet^\delta) + \sqrt{2k}(c_\star^\delta + c_?^\delta)}\ , \tag{24}$$

where for each construction, the parameters are given in Table 4 and in Table 2.

Similarly to what was done in the previous subsection, we compare the values for the three constructions, for different sizes of the alphabet, in Table 5.

ε-NFAs	ε-transitions
Thompson	(0,1,0,4,1,3,4)
SSS	(0,1,0,2,0,1,3)
ε-follow	(0,1,0,0,0,1,2)

Table 4: Characteristic constants for ε-transitions

k	2	10	50	100	∞	worst-case
Thompson	0.86	0.80	0.76	0.75	$5/7 \sim 0.71$	1
SSS	0.75	0.65	0.58	0.56	$1/2$	1
ε-follow	0.6	0.4	0.23	0.13	0	1

Table 5: Average vs. worst-case density of ε-transitions

8 Final Remarks

In addition to the simulations of different models for regular languages, the descriptional complexity of closed operations over regular languages has been extensively studied. One of the most studied complexity measures for a regular language is the number of states of its minimal deterministic finite automaton (state complexity of the language). The state complexity of an operation over languages is the complexity of the resulting language as a function of the complexities of its arguments. Other models of computation (e.g. nondeterministic automata, two-way automata, regular expressions, grammars, etc.), other measures (number of transitions, number of symbols, etc.) and other classes of languages (classes of sub-regular languages, context-free languages, recursive languages, etc.) have also been studied. However, there is a meager amount of results dealing with the average-case. For regular languages, the authors are only aware of the following works. The average state complexity of union, intersection and concatenation of languages over an unary alphabet were studied by Nicaud [Nic99]. Domaratzki [Dom02] extended these analyses to the $\frac{1}{2}(\cdot)$ operation. Gruber and Holzer [GH07] studied the average state and transition complexity of DFAs and NFAs accepting finite languages. Using the framework of analytic combinatorics, Bassino *et. al* [BGN10] studied the average state complexity of regular operations on finite languages.

An alternative approach to obtain average-case complexity values relies on the uniform random generation and enumeration of combinatorial objects, for which experimentation with large samples is tractable. The mathematical tools described in the present paper have been used in that context for the enumeration of models of computation such as finite automata [DKS02], initially connect deterministic finite automata [AMR07, BN07], and regular expressions generated by different grammars [EKSW05, LS05].

Lots of work remain to be done on evaluating average-case complexity of, for instance, determinization of finite automata and operations over regular languages. We hope that the exposition here made will make this kind of analysis more accessible to a wider range of researchers.

Acknowledgements We thank Markus Holzer for the challenge to study the asymptotic average-case descriptional complexity of the conversion of regular expressions to ε-automata that triggered this work.

A A Pedestrian Example

For the evaluation of the number of the symbols a in the words of the language given by $(\mathtt{ab}+\mathtt{c}+\mathtt{d})^\star$, let w_n denote the number of words of length n, as in Section 2, and a_n the desired number of a's in the words of size n.

Note that words of size n must end with a symbol b, c or d. In each of the last two cases, the corresponding number of a's is exactly the number of a's in words of size $n-1$. For the words in the first case, there is an a in the next-to-last position, and the total number of a's in the $n-2$ sized prefix is the number of a's occurring in words of size $n-2$. Thus

$$a_n = a_{n-2} + w_{n-2} + 2a_{n-1} \quad (n \geq 2). \tag{25}$$

Multiplying equality (25) by z^n, adding all the resulting equations, and denoting by $A(z)$ the generating function corresponding to $(a_n)_n$, one gets

$$A(z) - a_0 - a_1 z = z^2 A(z) + z^2 W(z) + 2z(A(z) - a_0). \tag{26}$$

Since $a_0 = a_1 = 0$, and using equation (3), one obtains

$$A(z) = \frac{z^2}{1-2z-z^2}\, W(z) = \frac{z^2}{(1-2z-z^2)^2}. \tag{27}$$

Letting $\rho = -1+\sqrt{2}$ and $\bar{\rho} = -1-\sqrt{2}$, the two roots of the denominator, one can write the above rational fraction as

$$\begin{aligned} A(z) &= \frac{z^2}{(z-\rho)^2(z-\bar{\rho})^2} = \\ &= -\frac{1}{8\sqrt{2}}\frac{z^2}{z-\rho} + \frac{1}{8}\frac{z^2}{(z-\rho)^2} + \frac{1}{8\sqrt{2}}\frac{z^2}{z-\bar{\rho}} + \frac{1}{8}\frac{z^2}{(z-\bar{\rho})^2}. \end{aligned} \tag{28}$$

Now, differentiating

$$\frac{1}{1-\frac{z}{\rho}} = \sum_{n\geq 0}\left(\frac{z}{\rho}\right)^n,$$

one can easily get

$$\frac{1}{(z-\rho)^2} = \sum_{n\geq 1}\frac{n}{\rho^2}\left(\frac{z}{\rho}\right)^{n-1} = \sum_{n\geq 0}\frac{(n+1)}{\rho^2}\left(\frac{z}{\rho}\right)^n.$$

And thus, using this and the corresponding formulas for $\bar{\rho}$, one gets

$$A(z) = \sum_{n\geq 0}\left(\frac{1}{8\sqrt{2}\rho^{n+1}} + \frac{n+1}{8\rho^{n+2}} - \frac{1}{8\sqrt{2}\bar{\rho}^{n+1}} + \frac{n+1}{8\bar{\rho}^{n+2}}\right) z^{n+2}.$$

Therefore

$$a_n = \frac{1}{8\sqrt{2}\rho^{n-1}} + \frac{n-1}{8\rho^n} - \frac{1}{8\sqrt{2}\bar{\rho}^{n-1}} + \frac{n-1}{8\bar{\rho}^n}, \quad \forall n \geq 2.$$

Observing that $|\bar{\rho}| > 1$ one concludes that

$$a_n = \left(\frac{\rho}{8\sqrt{2}} + \frac{n-1}{8}\right)\frac{1}{\rho^n} + o(1),$$

and thus

$$a_n \sim \frac{1}{8}\left(n - \frac{1}{\sqrt{2}}\right)\rho^{-n}. \tag{29}$$

This approximation for a_n gives rise to the same asymptotic limit as the one obtained in Section 6.

References

[AHU74] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[AMR07] M. Almeida, N. Moreira, and R. Reis. Enumeration and generation with a string automata representation. *Theoret. Comput. Sci.*, 387(2):93–102, 2007. Special issue "Selected papers of DCFS 2006".

[BGN10] Frédérique Bassino, Laura Giambruno, and Cyril Nicaud. The average state complexity of rational operations on finite languages. *Int. J. Found. Comput. Sci.*, 21(4):495–516, 2010.

[BMMR11a] Sabine Broda, António Machiavelo, Nelma Moreira, and Rogério Reis. The average transition complexity of Glushkov and partial derivative automata. In G. Mauri and A. Leporati, editors, *15th International Conference on Developments in Language Theory, DLT 2011. Proceedings*, volume 6795 of *LNCS*, pages 93–104, Milano, Italy, July 2011. Springer.

[BMMR11b] Sabine Broda, António Machiavelo, Nelma Moreira, and Rogério Reis. On the average state complexity of partial derivative automata. *International Journal of Foundations of Computer Science*, 22(7):1593–1606, 2011.

[BN07] Frédérique Bassino and Cyril Nicaud. Enumeration and random generation of accessible automata. *Theoret. Comput. Sci.*, 381(1-3):86–104, 2007.

[Brz10] Janusz A. Brzozowski. Quotient complexity of regular languages. *Journal of Automata, Languages and Combinatorics*, 15(1/2):71–89, 2010.

[Brz12] Janusz Brzozowski. In search of most complex regular languages. In N. Moreira and R. Reis, editors, *17th International Conference on Implementation and Application of Automata, CIAA 2012. Proocedings*, volume 7381 of *LNCS*, pages 5–24, Porto, Portugal, July 2012. Springer.

[DKS02] Michael Domaratzki, D. Kisman, and J. Shallit. On the number of distinct languages accepted by finite automata with n states. *J. of Automata, Languages and Combinatorics*, 7(4):469–486, 2002.

[Dom02] Michael Domaratzki. State complexity of proportional removals. *Journal of Automata, Languages and Combinatorics*, 7(4):455–468, 2002.

[EKSW05] Keith Ellul, Bryan Krawetz, Jeffrey Shallit, and Ming-Wei Wang. Regular expressions: New results and open problems. *J. Aut., Lang. and Combin.*, 10(4):407–437, 2005.

[FS08] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2008.

[GG10] Hermann Gruber and Stefan Gulan. Simplifying regular expressions. In Adrian Horia Dediu, Henning Fernau, and Carlos Martín-Vide, editors, *4th International Conference on Language and Automata Theory and Applications, LATA 2010. Proceedings*, volume 6031 of *LNCS*, pages 285–296, Trier, Germany, May 2010. Springer.

[GH07] Hermann Gruber and Markus Holzer. On the average state and transition complexity of finite languages. *Theor. Comput. Sci.*, 387(2):155–166, 2007.

[GKK+02] Jonathan Goldstine, Martin Kappes, Chandra M. R. Kintala, Hing Leung, Andreas Malcher, and Detlef Wotschke. Descriptional complexity of machines with limited resources. *J. UCS*, 8(2):193–234, 2002.

[HK10a] M. Holzer and M. Kutrib. The complexity of regular(-like) expressions. In Y. Gao, H. Lu, S.Seki, and S. Yu, editors, *14th International Conference on Developments in Language Theory, DLT 2010. Proceedings*, volume 6224 of *LNCS*, pages 16–30, London, Ontario, CA, August 2010. Springer.

[HK10b] Markus Holzer and Martin Kutrib. Descriptional complexity – An introductory survey. In Carlos Martín-Vide, editor, *Scientific Applications of Language Methods*, pages 1–58. World Scientific, 2010.

[HK11] Markus Holzer and Martin Kutrib. Descriptional and computational complexity of finite automata – A survey. *Inf. Comput.*, 209(3):456–470, 2011.

[Hro02] Juraj Hromkovic. Descriptional complexity of finite automata: Concepts and open problems. *Journal of Automata, Languages and Combinatorics*, 7(4):519–531, 2002.

[IY03] L. Ilie and S. Yu. Follow automata. *Inf. Comput.*, 186(1):140–162, 2003.

[Knu73a] Donald E. Knuth. *The Art of Computer Programming, Volume I: Fundamental Algorithms, 2nd Edition.* Addison-Wesley, 1973.

[Knu73b] Donald E. Knuth. *The Art of Computer Programming, Volume III: Sorting and Searching.* Addison-Wesley, 1973.

[Knu81] Donald E. Knuth. *The Art of Computer Programming, Volume II: Seminumerical Algorithms, 2nd Edition.* Addison-Wesley, 1981.

[LS05] Jonathan Lee and Jeffrey Shallit. Enumerating regular expressions and their languages. In M. Domaratzki, A. Okhotin, K. Salomaa, and S. Yu, editors, *9th International Conference on Implementation and Application of Automata, CIAA 2004. Proceedings*, volume 3314 of *LNCS*, pages 2–22. Springer, 2005.

[Nic99] Cyril Nicaud. Average state complexity of operations on unary automata. In M. Kurylowski, L. Pacholski, and T. Wierzbicki, editors, *24th Symposium, Mathematical Foundations of Computer Science, MFCS 1999. Proceedings*, volume 1672 of *LNCS*, pages 231–240. Springer, 1999.

[Nic09] Cyril Nicaud. On the average size of Glushkov's automata. In Adrian Horia Dediu, Armand-Mihai Ionescu, and Carlos Martín-Vide, editors, *3rd International Conference on Language and Automata Theory and Applications, LATA 2009. Proceedings*, volume 5457 of *LNCS*, pages 626–637. Springer, 2009.

[SF96] R. Sedgewick and P. Flajolet. *Analysis of Algorithms.* Addison Wesley, 1996.

[SSS90] S. Sippu and E. Soisalon-Soininen. *Parsing Theory*, volume II: LR(k) and LL(k) Parsing of *EATCS Monographs on Theoretical Computer Science.* Springer, 1990.

[Tho68] K. Thompson. Regular expression search algorithm. *Communications of the ACM*, 11(6):410–422, 1968.

[YG11] Sheng Yu and Yuan Gao. State complexity research and approximation. In G. Mauri and A. Leporati, editors, *15th International Conference on Developments in Language Theory, DLT 2011. Proceedings*, volume 6795 of *LNCS*, pages 46–57, Milano, Italy, July 2011.

[Yu05] Sheng Yu. State complexity: Recent results and open problems. *Fundam. Inform.*, 64(1-4):471–480, 2005.