Prefix and Right-Partial Derivative Automata ⋆

Eva Maia, Nelma Moreira, Rogério Reis

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

Abstract. Recently, Yamamoto presented a new method for the conversion from regular expressions (REs) to non-deterministic finite automata (NFA) based on the Thompson ε-NFA ($\mathcal{A}_{\mathsf{T}}$). The $\mathcal{A}_{\mathsf{T}}$ automaton has two quotients discussed: the suffix automaton $\mathcal{A}_{\mathsf{suf}}$ and the prefix automaton, $\mathcal{A}_{\mathsf{pre}}$. Eliminating ε-transitions in $\mathcal{A}_{\mathsf{T}}$, the Glushkov automaton ($\mathcal{A}_{\mathsf{pos}}$) is obtained. Thus, it is easy to see that $\mathcal{A}_{\mathsf{suf}}$ and the partial derivative automaton ($\mathcal{A}_{\mathsf{pd}}$) are the same. In this paper, we characterise the $\mathcal{A}_{\mathsf{pre}}$ automaton as a solution of a system of left RE equations and express it as a quotient of $\mathcal{A}_{\mathsf{pos}}$ by a specific left-invariant equivalence relation. We define and characterise the right-partial derivative automaton ($\overleftarrow{\mathcal{A}}_{\mathsf{pd}}$). Finally, we study the average size of all these constructions both experimentally and from an analytic combinatorics point of view.

1 Introduction

Conversion methods from regular expressions to equivalent nondeterministic finite automata have been widely studied. Resulting NFAs can have ε-transitions or not. The standard conversion with ε-transitions is the Thompson automaton ($\mathcal{A}_{\mathsf{T}}$) [15] and the standard conversion without ε-transitions is the Glushkov (or position) automaton ($\mathcal{A}_{\mathsf{pos}}$) [9]. Other conversions such as partial derivative automaton ($\mathcal{A}_{\mathsf{pd}}$) [1, 13] or follow automaton ($\mathcal{A}_{\mathsf{f}}$) [10] were proved to be quotients of the $\mathcal{A}_{\mathsf{pos}}$, by specific right-invariant equivalence relations [6, 10]. In particular, for REs under special conditions, $\mathcal{A}_{\mathsf{pd}}$ is an optimal conversion method [12]. Moreover, asymptotically and on average, the size of $\mathcal{A}_{\mathsf{pd}}$ is half the size of $\mathcal{A}_{\mathsf{pos}}$ [3]. Reductions on the size of NFAs using left-relations was studied recently by Ko and Han [11].

Yamamoto [16] presented a new conversion method based on the $\mathcal{A}_{\mathsf{T}}$. Given a $\mathcal{A}_{\mathsf{T}}$, two automata are constructed by merging $\mathcal{A}_{\mathsf{T}}$ states: in one, the suffix automaton ($\mathcal{A}_{\mathsf{suf}}$), states with the same right languages and in the other, the prefix automaton ($\mathcal{A}_{\mathsf{pre}}$), states with the same left languages. $\mathcal{A}_{\mathsf{suf}}$ corresponds to $\mathcal{A}_{\mathsf{pd}}$, which is not a surprise because it is known that if ε-transitions are eliminated from $\mathcal{A}_{\mathsf{T}}$, the $\mathcal{A}_{\mathsf{pos}}$ is obtained [8]. $\mathcal{A}_{\mathsf{pre}}$ is a quotient by a left-invariant

⋆ This work was partially funded by the European Regional Development Fund through the programme COMPETE and by the Portuguese Government through the FCT under project UID/MAT/00144/2013 and project FCOMP-01-0124-FEDER-020486. Eva Maia was also funded by FCT grant SFRH/BD/78392/2011.

relation. In this paper, we further study conversions from REs to NFAs based on left-invariant relations. Using the notion of right-partial derivatives introduced by Champarnaud *et. al* [4], we define the right-partial derivative automaton $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}$, characterise its relation with $\mathcal{A}_{\mathsf{pd}}$ and $\mathcal{A}_{\mathsf{pos}}$, and study its average size. We construct the $\mathcal{A}_{\mathsf{pre}}$ automaton directly from a regular expression without use the $\mathcal{A}_{\mathsf{T}}$ automaton, and we show that it is also a quotient of the $\mathcal{A}_{\mathsf{pos}}$. However, the experimental results suggest that, on average, the reduction on the size of the $\mathcal{A}_{\mathsf{pos}}$ is not large. Considering the framework of analytic combinatorics we study this reduction.

2 Regular Expressions and Automata

Given an alphabet $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ of size k, the set RE of *regular expressions* α over Σ is defined by the following grammar:

$$\alpha := \emptyset \mid \varepsilon \mid \sigma_1 \mid \cdots \mid \sigma_k \mid (\alpha + \alpha) \mid (\alpha \cdot \alpha) \mid (\alpha)^\star, \tag{1}$$

where the $\cdot$ is often omitted. If two REs α and β are syntactically equal, we write $\alpha \sim \beta$. The *size* of a RE α, $|\alpha|$, is its number of symbols, disregarding parenthesis, and its *alphabetic size*, $|\alpha|_\Sigma$, is the number of occurrences of letters from Σ. A RE α is *linear* if all its letters occurs only once. The language represented by a RE α is denoted by $\mathcal{L}(\alpha)$. Two REs α and β are *equivalent* if $\mathcal{L}(\alpha) = \mathcal{L}(\beta)$, and we write $\alpha = \beta$. We define the function ε by $\varepsilon(\alpha) = \varepsilon$ if $\varepsilon \in \mathcal{L}(\alpha)$ and $\varepsilon(\alpha) = \emptyset$, otherwise. This function can be naturally extended to sets of REs and languages. We consider REs reduced by the following rules: $\varepsilon\alpha = \alpha = \alpha\varepsilon$, $\emptyset + \alpha = \alpha = \alpha + \emptyset$, and $\emptyset\alpha = \emptyset = \alpha\emptyset$. Given a language $\mathcal{L} \subseteq \Sigma^\star$ and a word $w \in \Sigma^\star$, the *left quotient* of $\mathcal{L}$ w.r.t. w is the language $w^{-1}L = \{x \mid wx \in L\}$, and the *right quotient* of $\mathcal{L}$ w.r.t. w is the language $\mathcal{L}w^{-1} = \{x \mid xw \in \mathcal{L}\}$. The *reversal* of a word $w = \sigma_1\sigma_2\cdots\sigma_n$ is $w^R = \sigma_n\cdots\sigma_2\sigma_1$. The *reversal* of a language $\mathcal{L}$, denoted by $\mathcal{L}^R$, is the set of words whose reversal is on $\mathcal{L}$. The reversal of $\alpha \in \mathsf{RE}$ is denoted by α^R. The reversal of set of REs is the set of the reversal of its elements. It is not difficult to verify that $\mathcal{L}w^{-1} = ((w^R)^{-1}L^R)^R$.

A *nondeterministic finite automaton* (NFA) is a five-tuple $A = (Q, \Sigma, \delta, I, F)$ where Q is a finite set of states, Σ is a finite alphabet, $I \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of final states, and $\delta : Q \times \Sigma \to 2^Q$ is the transition function. The transition function can be extended to words and to sets of states in the natural way. When $I = \{q_0\}$, we use $I = q_0$. Given a state $q \in Q$, the *right language* of q is $\mathcal{L}_q(A) = \{w \in \Sigma^\star \mid \delta(q, w) \cap F \neq \emptyset\}$, and the *left language* is $\overleftarrow{\mathcal{L}}_q(A) = \{w \in \Sigma^\star \mid q \in \delta(I, w)\}$. The language accepted by A is $\mathcal{L}(A) = \bigcup_{q \in I} \mathcal{L}_q(A)$. Two NFAs are *equivalent* if they accept the same language. If two NFAs A and B are isomorphic, we write $A \simeq B$. An NFA is *deterministic* if for all $(q, \sigma) \in Q \times \Sigma$, $|\delta(q, \sigma)| \leq 1$ and $|I| = 1$. The *reversal* of an automaton A is the automaton A^R, where the sets of initial and final states are swapped and all transitions are reversed. Given an equivalence relation $\equiv$ in Q, the *quotient automaton* $A/_{\equiv} = (Q/_{\equiv}, \Sigma, \delta/_{\equiv}, I/_{\equiv}, F/_{\equiv})$ is defined in the usual way. A relation $\equiv$ is *right invariant* w.r.t. A if and only if: $\equiv \subseteq (Q-F)^2 \cup F^2$ and $\forall p,q \in Q, \sigma \in \Sigma$, if $p \equiv q$, then ${}^{\delta(p,\sigma)}/_{\equiv} = {}^{\delta(q,\sigma)}/_{\equiv}$. A relation $\equiv$ is a *left invariant* relation w.r.t. A if and only if it is a right-invariant relation w.r.t. A^R.

The right languages $\mathcal{L}_i$, for $i \in Q = [0,n]$, define a system of right equations, $\mathcal{L}_i = \bigcup_{j=1}^{k} \sigma_j \left(\bigcup_{m\in I_{ij}} \mathcal{L}_m\right) \cup \varepsilon(\mathcal{L}_i)$, where $I_{ij} \subseteq [0,n]$, $m \in I_{ij} \Leftrightarrow m \in \delta(i,\sigma_j)$, and $\mathcal{L}(A) = \bigcup_{i\in I} \mathcal{L}_i$. In the same manner, the left languages of the states of A define a system of left equations $\overleftarrow{\mathcal{L}}_i = \bigcup_{j=1}^{k} \left(\bigcup_{m\in I_{ij}} \overleftarrow{\mathcal{L}}_m\right) \sigma_j \cup \varepsilon(\overleftarrow{\mathcal{L}}_i)$, where $I_{ij} \subseteq [0,n]$, $m \in I_{ij} \Leftrightarrow i \in \delta(m,\sigma_j)$, and $\mathcal{L}(A) = \bigcup_{i\in F} \overleftarrow{\mathcal{L}}_i$.

2.1 Glushkov and Partial Derivative Automata

In the following we review two constructions which define NFAs equivalent to a given regular expression $\alpha \in \mathsf{RE}$. Let $\mathsf{pos}(\alpha) = \{1,2,\ldots,|\alpha|_\Sigma\}$ be the set of letter positions in α, and let $\mathsf{pos}_0(\alpha) = \mathsf{pos}(\alpha) \cup \{0\}$. We consider the expression $\overline{\alpha}$ obtained by marking each letter with its position in α, i.e. $\mathcal{L}(\overline{\alpha}) \in \overline{\Sigma}^\star$ where $\overline{\Sigma} = \{\sigma_i \mid \sigma \in \Sigma, 1 \leq i \leq |\alpha|_\Sigma\}$. The same notation is used to remove the markings, i.e., $\overline{\overline{\alpha}} = \alpha$. For $\alpha \in \mathsf{RE}$ and $i \in \mathsf{pos}(\alpha)$, let $\mathsf{first}(\alpha) = \{i \mid \exists w \in \overline{\Sigma}^\star, \sigma_i w \in \mathcal{L}(\overline{\alpha})\}$, $\mathsf{last}(\alpha) = \{i \mid \exists w \in \overline{\Sigma}^\star, w\sigma_i \in \mathcal{L}(\overline{\alpha})\}$ and $\mathsf{follow}(\alpha,i) = \{j \mid \exists u,v \in \overline{\Sigma}^\star, u\sigma_i\sigma_j v \in \mathcal{L}(\overline{\alpha})\}$. The *Glushkov automaton* (or position automaton) for α is $\mathcal{A}_{\mathsf{pos}}(\alpha) = (\mathsf{pos}_0(\alpha), \Sigma, \delta_{\mathsf{pos}}, 0, F)$, 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 $\varepsilon(\alpha) = \varepsilon$, and $F = \mathsf{last}(\alpha)$, otherwise. We note that the number of states of $\mathcal{A}_{\mathsf{pos}}(\alpha)$ is exactly $|\alpha|_\Sigma + 1$.

The partial derivative automaton of a regular expression was introduced independently by Mirkin [13] and Antimirov [1]. Champarnaud and Ziadi [5] proved that the two formulations are equivalent. For a regular expression $\alpha \in \mathsf{RE}$ and a symbol $\sigma \in \Sigma$, the set of left-partial derivatives of α w.r.t. σ is defined inductively as follows:

$$\begin{array}{ll} \partial_\sigma(\emptyset) \ = \partial_\sigma(\varepsilon) = \emptyset & \partial_\sigma(\alpha+\beta) = \partial_\sigma(\alpha) \cup \partial_\sigma(\beta) \\ \partial_\sigma(\sigma') = \begin{cases} \{\varepsilon\} \text{ if } \sigma' = \sigma \\ \emptyset \quad \text{otherwise} \end{cases} & \begin{array}{l} \partial_\sigma(\alpha\beta) \ \ \ = \partial_\sigma(\alpha)\beta \cup \varepsilon(\alpha)\partial_\sigma(\beta) \\ \partial_\sigma(\alpha^\star) \ \ \ = \partial_\sigma(\alpha)\alpha^\star \end{array} \end{array} \qquad (2)$$

where for any $S \subseteq \mathsf{RE}$, $S\emptyset = \emptyset S = \emptyset$, $S\varepsilon = \varepsilon S = S$, and $S\beta = \{\alpha\beta | \alpha \in S\}$ if $\beta \neq \emptyset, \varepsilon$ (and analogously for βS). The definition of left-partial derivatives can be extended in a natural way to sets of regular expressions, words, and languages. We have that $w^{-1}\mathcal{L}(\alpha) = \mathcal{L}(\partial_w(\alpha)) = \bigcup_{\tau\in\partial_w(\alpha)} \mathcal{L}(\tau)$, for $w \in \Sigma^\star$. The set of all partial derivatives of α w.r.t. words is denoted by $\mathsf{PD}(\alpha) = \partial_{\Sigma^\star}(\alpha)$. The *partial derivative automaton* of α is $\mathcal{A}_{\mathsf{pd}}(\alpha) = (\mathsf{PD}(\alpha), \Sigma, \delta_{pd}, \alpha, F_{pd})$, where $\delta_{pd} = \{(\tau,\sigma,\tau') \mid \tau \in \mathsf{PD}(\alpha),\ \sigma \in \Sigma,\ \tau' \in \partial_\sigma(\tau)\}$ and $F_{pd} = \{\tau \in \mathsf{PD}(\alpha) \mid \varepsilon(\tau) = \varepsilon\}$.

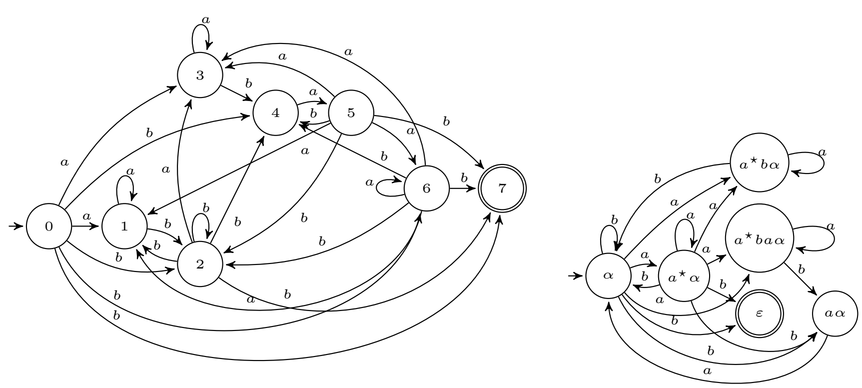
As noted by Broda et al. [3] and Maia et al. [12], following Mirkin's construction, the partial derivative automaton of α can be inductively constructed. A *(right) support* for α is a set of regular expressions $\{\alpha_1,\ldots,\alpha_n\}$ such that $\alpha_i = \sigma_1\alpha_{i1} + \cdots + \sigma_k\alpha_{ik} + \varepsilon(\alpha_i)$, $i \in [0,n]$, $\alpha_0 \sim \alpha$ and α_{ij} is a linear combination of α_l, $l \in [1,n]$ and $j \in [1,k]$. The set $\pi(\alpha)$ inductively defined below is a right support of α.

$$\begin{array}{lr} \pi(\emptyset) = \emptyset & \pi(\alpha+\beta) = \pi(\alpha) \cup \pi(\beta) \\ \pi(\varepsilon) = \emptyset & \pi(\alpha\beta) = \pi(\alpha)\beta \cup \pi(\beta) \\ \pi(\sigma) = \{\varepsilon\} & \pi(\alpha^\star) = \pi(\alpha)\alpha^\star. \end{array} \tag{3}$$

Champarnaud and Ziadi proved that $\mathsf{PD}(\alpha) = \pi(\alpha) \cup \{\alpha\}$ and the transition function of $\mathcal{A}_{\mathsf{pd}}$ can also be defined inductively from the system of equations above. Let $\varphi(\alpha) = \{(\sigma, \gamma) \mid \gamma \in \partial_\sigma(\alpha), \sigma \in \Sigma\}$ and $\lambda(\alpha) = \{\alpha' \mid \alpha' \in \pi(\alpha),\ \varepsilon(\alpha') = \varepsilon\}$, where both sets can be inductively defined using (2) and (3). We have, $\delta_{pd} = \{\alpha\} \times \varphi(\alpha) \cup F(\alpha)$ where the result of the $\times$ operation is seen as a set of triples and the set F is defined inductively by:

$$\begin{array}{r} F(\emptyset) = F(\varepsilon) = F(\sigma) = \emptyset,\ \sigma \in \Sigma \\ F(\alpha+\beta) = F(\alpha) \cup F(\beta) \\ F(\alpha\beta) = F(\alpha)\beta \cup F(\beta) \cup \lambda(\alpha)\beta \times \varphi(\beta) \\ F(\alpha^\star) = F(\alpha)\alpha^\star \cup (\lambda(\alpha) \times \varphi(\alpha))\alpha^\star. \end{array} \tag{4}$$

Note that the concatenation of a transition (α, σ, β) with a RE γ is defined by $(\alpha, \sigma, \beta)\gamma = (\alpha\gamma, \sigma, \beta\gamma)$ (similarly $\gamma(\alpha, \sigma, \beta) = (\gamma\alpha, \sigma, \gamma\beta)$), if $\gamma \notin \{\emptyset, \varepsilon\}$, $(\alpha, \sigma, \beta)\emptyset = \emptyset$ and $(\alpha, \sigma, \beta)\varepsilon = (\alpha, \sigma, \beta)$. Then, $\mathcal{A}_{\mathsf{pd}}(\alpha) = (\pi(\alpha) \cup \{\alpha\}, \Sigma, \{\alpha\} \times \varphi(\alpha) \cup F(\alpha), \alpha, \lambda(\alpha) \cup \varepsilon(\alpha)\{\alpha\})$. In Fig. 1 are represented $\mathcal{A}_{\mathsf{pos}}(\alpha)$ and $\mathcal{A}_{\mathsf{pd}}(\alpha)$, where $\alpha = \beta b$ and $\beta = (a^\star b + a^\star ba + a^\star)^\star$.



(a) $\mathcal{A}_{\mathsf{pos}}(\alpha)$ (b) $\mathcal{A}_{\mathsf{pd}}(\beta b)$

Fig. 1. Automata for $\alpha = \beta b$ with $\beta = (a^\star b + a^\star ba + a^\star)^\star$.

Champarnaud and Ziadi [6] showed that the partial derivative automaton is a quotient of the Glushkov automaton by the right-invariant equivalence relation $\equiv_c$, such that $i \equiv_c j$ if $\partial_{w\sigma_i}(\overline{\alpha}) = \partial_{w\sigma_j}(\overline{\alpha})$, for $i, j \in \mathsf{pos}_0(\alpha)$ and let $\sigma_0 = \varepsilon$. It is known that $\partial_{w\sigma_i}(\overline{\alpha})$ is either empty or an unique singleton for all $w \in \overline{\Sigma}^\star$.

3 Right-Partial Derivative Automata

The concept of right-partial derivative was introduced by Champarnaud *et. al.* For a regular expression $\alpha \in \mathsf{RE}$ and a symbol $\sigma \in \Sigma$, the set of right-partial derivatives of α w.r.t. σ, $\overleftarrow{\partial}_\sigma(\alpha)$, is defined in the same way as the set of left-partial derivatives except for the following two rules:

$$\overleftarrow{\partial}_\sigma(\alpha\beta) = \alpha\overleftarrow{\partial}_\sigma(\beta) \cup \varepsilon(\beta)\overleftarrow{\partial}_\sigma(\alpha) \qquad \overleftarrow{\partial}_\sigma(\alpha^\star) = \alpha^\star\overleftarrow{\partial}_\sigma(\alpha). \tag{5}$$

This definition can be extended in a natural way to sets of regular expressions, words, and languages. The set of all right-partial derivatives of α w.r.t. words is denoted by $\overleftarrow{\mathsf{PD}}(\alpha) = \overleftarrow{\partial}_{\Sigma^\star}(\alpha)$. The right- and left-partial derivatives of α w.r.t. $w \in \Sigma^\star$ are related by $\overleftarrow{\partial}_w(\alpha) = (\partial_{w^R}(\alpha^R))^R$. Thus, $\mathcal{L}(\overleftarrow{\partial}_w(\alpha)) = \mathcal{L}(\alpha)w^{-1}$. The right-partial derivative automaton of α, $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}(\alpha)$, can be defined inductively as a solution of a left system of expression equations, $\alpha_i = \alpha_{i1}\sigma_1 + \cdots + \alpha_{ik}\sigma_k + \varepsilon(\alpha_i)$, $i \in [0, n]$, $\alpha_0 \sim \alpha$, α_{ij} is a linear combination of α_l, $l \in [1, n]$ and $j \in [1, k]$.

Proposition 1. *The set of regular expressions $\overleftarrow{\pi}(\alpha)$ defined in the same way as the set π, except for the concatenation and Kleene star rules, is a solution of a left system of expression equations,*

$$\overleftarrow{\pi}(\alpha\beta) = \alpha\overleftarrow{\pi}(\beta) \cup \overleftarrow{\pi}(\alpha) \qquad \overleftarrow{\pi}(\alpha^\star) = \alpha^\star\overleftarrow{\pi}(\alpha). \tag{6}$$

Again, the solution of the system of equations also allows to inductively define the transition function. Let $\overleftarrow{\varphi}(\alpha) = \{(\gamma, \sigma) \mid \gamma \in \overleftarrow{\partial}_\sigma(\alpha), \sigma \in \Sigma\}$ and $\overleftarrow{\lambda}(\alpha) = \{\alpha' \mid \alpha' \in \overleftarrow{\pi}(\alpha),\ \varepsilon(\alpha') = \varepsilon\}$, where both sets can be inductively defined using (5) and (6). The set of transitions of $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}(\alpha)$ is $\overleftarrow{\varphi}(\alpha) \times \{\alpha\} \cup \overleftarrow{F}(\alpha)$ and the set $\overleftarrow{F}(\alpha)$ is defined similarly to the set $F(\alpha)$ except for the two following rules:

$$\begin{aligned}\overleftarrow{F}(\alpha\beta) &= \alpha\overleftarrow{F}(\beta) \cup \overleftarrow{F}(\alpha) \cup \varphi(\alpha) \times (\alpha\overleftarrow{\lambda}(\beta))\\ \overleftarrow{F}(\alpha^\star) &= \alpha^\star\overleftarrow{F}(\alpha) \cup \alpha^\star(\overleftarrow{\varphi}(\alpha) \times \overleftarrow{\lambda}(\alpha)).\end{aligned} \tag{7}$$

The *right-partial derivative automaton* of α is $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}(\alpha) = (\overleftarrow{\pi}(\alpha) \cup \{\alpha\}, \Sigma, \overleftarrow{\varphi}(\alpha) \times \{\alpha\} \cup \overleftarrow{F}(\alpha), \overleftarrow{\lambda}(\alpha) \cup \varepsilon(\alpha)\{\alpha\}, \{\alpha\})$. In Fig. 3(a) is represented the $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}$ of the RE βb considered in Fig. 1. Note that the sizes of $\pi(\alpha)$ and $\overleftarrow{\pi}(\alpha)$ are not comparable in general. For example, $|\pi(\beta b)| > |\overleftarrow{\pi}(\beta b)|$, but if we consider $\alpha = b(ba^\star + aba^\star + a^\star)^\star$ then $|\pi(\alpha)| < |\overleftarrow{\pi}(\alpha)|$. The following result relates the functions defined above to the ones used to define the $\mathcal{A}_{\mathsf{pd}}$ is given by the following result.

Proposition 2. *Let α be a regular expression. Then $\overleftarrow{\pi}(\alpha) = (\pi(\alpha^R))^R$, $\overleftarrow{\lambda}(\alpha) = (\lambda(\alpha^R))^R$, $\overleftarrow{\varphi}(\alpha) = (\varphi(\alpha^R))^R$ and $\overleftarrow{F}(\alpha) = (F(\alpha^R))^R$.*

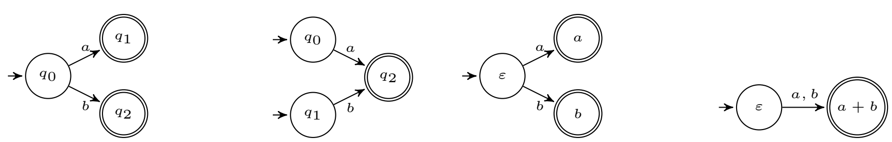
From the previous result and the fact that $\mathcal{A}_{\mathsf{pd}}(\alpha) \simeq {\mathcal{A}_{\mathsf{pos}}(\alpha)}/_{\equiv_c}$ we have

Proposition 3. *For any $\alpha \in \mathsf{RE}$,*

1. $(\mathcal{A}_{\mathsf{pd}}(\alpha^R))^R \simeq \overleftarrow{\mathcal{A}}_{\mathsf{pd}}(\alpha)$.
2. $\mathcal{L}(\overleftarrow{\mathcal{A}}_{\mathsf{pd}}(\alpha)) = \mathcal{L}(\alpha)$.
3. $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}(\alpha) \simeq {(\mathcal{A}_{\mathsf{pos}}(\alpha^R))^R}/_{\equiv_c}$.

4 Prefix Automata

Yamamoto [16] presented a new algorithm for converting a regular expression into an equivalent NFA. First, a labeled version of the usual Thompson NFA $(Q, \Sigma, \delta, I, F)$ is obtained, where each state q is labeled with two regular expressions, one that corresponds to its left language, $LP(q)$, and the other to its right language, $LS(q)$. States which in-transitions are labeled with a letter are called *sym-states*. Then the equivalence relations $\equiv_{pre}$ and $\equiv_{suf}$ are defined on the set of sym-states: for two states $p, q \in Q$, $p \equiv_{pre} q$ if and only if $LP(p) = LP(q)$; and $p \equiv_{suf} q$ if and only if $LS(p) = LS(q)$. The *prefix automaton* $\mathcal{A}_{\mathsf{pre}}$ and the *suffix automaton* $\mathcal{A}_{\mathsf{suf}}$ are the quotient automata by these relations. The final automaton is a combination of these two. The author also shows that $\mathcal{A}_{\mathsf{suf}}$ coincides with $\mathcal{A}_{\mathsf{pd}}$. This relation between $\mathcal{A}_{\mathsf{pd}}$ and $\mathcal{A}_{\mathsf{suf}}$ could lead us to think that $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}$ coincide with $\mathcal{A}_{\mathsf{pre}}$, which is not true. For instance, considering $\alpha = a+b$, the $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}(\alpha)$ has 2 states and the $\mathcal{A}_{\mathsf{pre}}(\alpha)$ has 3 states (see Fig. 2). Note that both automata are obtained from another automaton by merging the states with the same left language: while the $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}(\alpha)$ is obtained from $(\mathcal{A}_{\mathsf{pos}}(\alpha^R))^R$, we will see that the $\mathcal{A}_{\mathsf{pre}}(\alpha)$ is obtained from $\mathcal{A}_{\mathsf{pos}}(\alpha)$.



(a) $\mathcal{A}_{\mathsf{pos}}(\alpha)$ (b) $(\mathcal{A}_{\mathsf{pos}}(\alpha^R))^R$ (c) $\mathcal{A}_{\mathsf{pre}}(\alpha)$ (d) $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}(\alpha)$

Fig. 2. Automata for $\alpha = a + b$.

The LP labelling scheme proposed by Yamamoto can be obtained as a solution of a system of expression equations for a RE α, as done both for $\mathcal{A}_{\mathsf{pd}}$ and $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}$. Consider a system of left equations $\alpha_i = \alpha_{i1}\sigma_1 + \cdots + \alpha_{ik}\sigma_k$, $i \in [1, n]$, where $\alpha = \sum_{i \in I \subseteq [0,n]} \alpha_i$, $\alpha_{ij} = \sum_{l \in I_{ij} \subseteq [0,n]} \alpha_l$ and $\alpha_0 \sim \varepsilon$.

Proposition 4. *The set* $\mathsf{Pre}(\alpha)$ *inductively defined as follows:*

$$\begin{array}{ll} \mathsf{Pre}(\emptyset) = \emptyset & \mathsf{Pre}(\alpha + \beta) = \mathsf{Pre}(\alpha) \cup \mathsf{Pre}(\beta) \\ \mathsf{Pre}(\varepsilon) = \emptyset & \mathsf{Pre}(\alpha\beta) = \alpha\mathsf{Pre}(\beta) \cup \mathsf{Pre}(\alpha) \\ \mathsf{Pre}(\sigma) = \{\sigma\} & \mathsf{Pre}(\alpha^\star) = \alpha^\star\mathsf{Pre}(\alpha). \end{array} \tag{8}$$

is a solution (left support) of the system of left equations defined above.

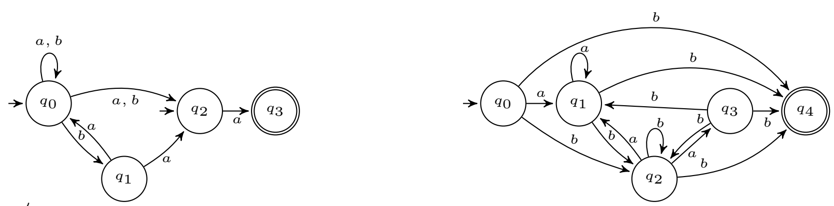
The set $\mathsf{Pre}_0(\alpha) = \mathsf{Pre}(\alpha) \cup \{\varepsilon\}$ constitutes the set of states of the prefix automaton $\mathcal{A}_{\mathsf{pre}}(\alpha)$. It also follows from the resolution of the above system of equations, that the set of transitions of $\mathcal{A}_{\mathsf{pre}}(\alpha)$ can be inductively defined. Let $\mathsf{P}(\alpha)$, $\psi(\alpha)$, and $\mathsf{T}(\alpha)$ be defined, respectively, as follows:

$$\begin{array}{ll} \mathsf{P}(\emptyset) = \emptyset & \mathsf{P}(\alpha + \beta) = \mathsf{P}(\alpha) \cup \mathsf{P}(\beta) \\ \mathsf{P}(\varepsilon) = \{\varepsilon\} & \mathsf{P}(\alpha\beta) = \alpha\mathsf{P}(\beta) \cup \varepsilon(\beta)\mathsf{P}(\alpha) \\ \mathsf{P}(\sigma) = \{\sigma\} & \mathsf{P}(\alpha^\star) = \alpha^\star\mathsf{P}(\alpha). \end{array} \tag{9}$$

$$\begin{array}{ll} \psi(\emptyset) = \emptyset & \psi(\alpha+\beta) = \psi(\alpha) \cup \psi(\alpha) \\ \psi(\varepsilon) = \emptyset & \psi(\alpha\beta) = \psi(\alpha) \cup \varepsilon(\alpha)\ \alpha\ \psi(\beta) \\ \psi(\sigma) = \{(\sigma,\sigma)\} & \psi(\alpha^\star) = \alpha^\star \psi(\alpha) \end{array} \tag{10}$$

$$\begin{aligned} \mathsf{T}(\emptyset) = \mathsf{T}(\varepsilon) = \mathsf{T}(\sigma) &= \emptyset,\ \sigma \in \Sigma \\ \mathsf{T}(\alpha+\beta) &= \mathsf{T}(\alpha) \cup \mathsf{T}(\beta) \\ \mathsf{T}(\alpha\beta) &= \mathsf{T}(\alpha) \cup \alpha\mathsf{T}(\beta) \cup \mathsf{P}(\alpha) \times (\alpha\psi(\beta)) \\ \mathsf{T}(\alpha^\star) &= \alpha^\star\mathsf{T}(\alpha) \cup \alpha^\star(\mathsf{P}(\alpha) \times \psi(\alpha)). \end{aligned} \tag{11}$$

Therefore, $\mathcal{A}_{\mathsf{pre}}(\alpha) = (\mathsf{Pre}_0(\alpha), \Sigma, \{\varepsilon\}\times\psi(\alpha)\cup\mathsf{T}(\alpha), \varepsilon, \mathsf{P}(\alpha)\cup\varepsilon(\alpha))$. In Fig.3(b) we can see the $\mathcal{A}_{\mathsf{pre}}(\beta b)$, where the RE βb is the one of Fig. 1. From both figures we observe that $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}(\beta b)$ is the smallest of the four automaton constructions. We



(a) $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}(\beta b) : q_0 = \beta a^\star$, $q_1 = \beta a^\star b$, $q_2 = \beta$, $q_3 = \beta b$

(b) $\mathcal{A}_{\mathsf{pre}}(\beta b) : q_0 = \varepsilon$, $q_1 = \beta(a^\star a)$, $q_2 = \beta(a^\star b)$, $q_3 = \beta((a^\star b)a)$, $q_4 = \beta b$.

Fig. 3. Automata for βb, where $\beta = (a^\star b + a^\star ba + a^\star)^\star$

now show that the $\mathcal{A}_{\mathsf{pre}}$ is a quotient of $\mathcal{A}_{\mathsf{pos}}$. If α is a linear regular expression, $\mathcal{A}_{\mathsf{pos}}(\alpha)$ is deterministic and thus all its states have distinct left languages. Therefore, in this case, $\mathcal{A}_{\mathsf{pre}}(\alpha)$ coincides with $\mathcal{A}_{\mathsf{pos}}(\alpha)$ and $|\mathsf{Pre}(\alpha)| = |\alpha|_\Sigma$. For an arbitrary RE α, $\mathcal{A}_{\mathsf{pre}}(\overline{\alpha}) \simeq \mathcal{A}_{\mathsf{pos}}(\overline{\alpha})$. Let $\equiv_l$ be the equivalence relation in $\mathsf{Pre}(\overline{\alpha})$ such that for any regular expression α, $\forall \alpha_1, \alpha_2 \in \mathsf{Pre}(\overline{\alpha}),\ \alpha_1 \equiv_l \alpha_2 \Leftrightarrow \overline{\alpha_1} = \overline{\alpha_2}$. It is not difficult to see that $\equiv_l$ is a left-invariant relation.

Proposition 5. *Let α be a regular expression. Then $\mathcal{A}_{\mathsf{pre}}(\alpha) \simeq {}^{\mathcal{A}_{\mathsf{pos}}(\alpha)}\!/_{\equiv_l}$.*

By construction, the Glushkov automaton is homogeneous, i.e. the in- transitions of each state are all labelled by the same letter. It follows from Proposition 5 that this property also holds for $\mathcal{A}_{\mathsf{pre}}$.

5 Average-Case Complexity

We conducted some experimental tests in order to compare the sizes of $\mathcal{A}_{\mathsf{pos}}$, $\mathcal{A}_{\mathsf{pd}}$, $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}$ and $\mathcal{A}_{\mathsf{pre}}$ automata. We used the FAdo library[1] that includes implementations of the NFA conversions and also several tools for uniformly random generate regular expressions. In order to obtain regular expressions uniformly

[1] http://fado.dcc.fc.up.pt

generated in the size of the syntactic tree, a prefix notation version of the grammar was used. For each alphabet size, k, and $|\alpha|$, samples of 10 000 REs were generated, which is sufficient to ensure a 95% confidence level within a 1% error margin. Table 1 presents the average values obtained for $|\alpha| \in \{100, 500, 1000\}$ and $k \in \{2, 10\}$. These experiments suggest that in pratice the $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}$ and the

k	$\lvert\alpha\rvert$	$\lvert\mathsf{pos}_0\rvert$	$\lvert\delta_{pos}\rvert$	$\lvert\mathsf{PD}\rvert$	$\lvert\delta_\pi\rvert$	$\frac{\lvert\pi\rvert}{\lvert\mathsf{pos}\rvert}$	$\lvert\overleftarrow{\mathsf{PD}}\rvert$	$\lvert\delta_{\overleftarrow{\pi}}\rvert$	$\frac{\lvert\overleftarrow{\pi}\rvert}{\lvert\mathsf{pos}\rvert}$	$\lvert\mathsf{Pre}_0\rvert$	$\lvert\delta_{pre}\rvert$	$\frac{\lvert\mathsf{Pre}\rvert}{\lvert\mathsf{pos}\rvert}$	$1-\eta_k$
2	100	28.9	167.5	15.7	56.0	0.55	15.9	56.4	0.55	20.1	73.7	0.71	0.90
	500	139.9	1486.5	71.6	389.8	0.51	71.5	393.1	0.51	91.9	530.8	0.66	
10	100	42.5	159.4	23.8	73.7	0.56	23.8	72.9	0.56	38.5	130.4	0.91	0.99
	500	207.1	1019.1	113.2	423.8	0.55	112.4	425.6	0.54	186	807.1	0.90	
	1000	412.1	2182.1	223.7	884.1	0.54	223.1	884.5	0.54	369.5	1717.6	0.90	

Table 1. Experimental results for uniform random generated regular expressions.

$\mathcal{A}_{\mathsf{pd}}$ have the same size and the $\mathcal{A}_{\mathsf{pre}}$ is not significantly smaller then the $\mathcal{A}_{\mathsf{pos}}$. By Proposition 3, $|\alpha^R|_\Sigma = |\alpha|_\Sigma$ and by the fact that $\varepsilon \in \pi(\alpha)$ if and only if $\varepsilon \in \overleftarrow{\pi}(\alpha)$, the analysis of the average size of $\mathcal{A}_{\mathsf{pd}}(\alpha)$ presented in Broda *et al* [2] carries on to $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}(\alpha)$. Thus the average sizes of $\mathcal{A}_{\mathsf{pd}}$ and $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}$ are asymptotically the same. However, $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}(\alpha)$ has only one final state and its number of initial states is the number of final states of $\mathcal{A}_{\mathsf{pd}}(\alpha^R)$. As studied by Nicaud [14], the size of $\mathsf{last}(\alpha)$ tends asymptotically to a constant depending on k and $|\lambda(\alpha)|$ is half that size [3]. Thus, that constant value will be also the number of initial states of $\overleftarrow{\mathcal{A}}_{\mathsf{pd}}$. Following, again, the ideas in Broda *et al.*, we estimate the number of mergings of states that arise when computing $\mathcal{A}_{\mathsf{pre}}$ from $\mathcal{A}_{\mathsf{pos}}$. The $\mathcal{A}_{\mathsf{pre}}$ has at most $|\alpha|_\Sigma + 1$ states and this only occurs when all unions in $\mathsf{Pre}(\alpha)$ are disjoint. However there are cases in which this does not happen. For instance, when $\sigma \in \mathsf{Pre}(\beta) \cap \mathsf{Pre}(\gamma)$, then $|\mathsf{Pre}(\beta+\gamma)| = |\mathsf{Pre}(\beta) \cup \mathsf{Pre}(\gamma)| \leq |\mathsf{Pre}(\beta)| + |\mathsf{Pre}(\gamma)| - 1$ and $|\mathsf{Pre}(\beta^\star\gamma)| = |\beta^\star \mathsf{Pre}(\gamma) \cup \beta^\star \mathsf{Pre}(\beta)| \leq |\mathsf{Pre}(\beta)| + |\mathsf{Pre}(\gamma)| - 1$. In what follows we estimate the number of these non-disjoint unions, which correspond to a lower bound for the number of states merged in the $\mathcal{A}_{\mathsf{pos}}$ automaton. This is done in the framework of analytic combinatorics as expounded by Flajolet and Sedgewick [7]. The methods apply to generating functions $A(z) = \sum_n a_n z^n$ for a combinatorial class $\mathcal{A}$ with a_n objects of size n, denoted by $[z^n]A(z)$, and also bivariate functions $C(u, z) = \sum_\alpha u^{c(\alpha)} z^{|\alpha|}$, where $c(\alpha)$ is some measure of the object $\alpha \in \mathcal{A}$.

The regular expressions α_σ for which $\sigma \in \mathsf{Pre}(\alpha_\sigma)$, $\sigma \in \Sigma$, are generated by following grammar:

$$\alpha_\sigma := \sigma \mid \alpha_\sigma + \alpha \mid \alpha_{\overline{\sigma}} + \alpha_\sigma \mid \alpha_\sigma \cdot \alpha \mid \varepsilon \cdot \alpha_\sigma \tag{12}$$

The regular expressions that are not generated by α_σ are denoted by $\alpha_{\overline{\sigma}}$. The generating function for α_σ, $R_{\sigma,k}(z)$ satisfies

$$\begin{aligned} R_{\sigma,k}(z) =& z + zR_{\sigma,k}(z)R_k(z) + z(R_k(z) - R_{\sigma,k}(z))R_{\sigma,k}(z) + \\ &+ zR_{\sigma,k}(z)R_k(z) + z^2 R_{\sigma,k}(z) \end{aligned}$$

From this one gets

$$R_{\sigma,k}(z) = \frac{(z^2 + 3zR_k(z) - 1) + \sqrt{(z^2 + 3zR_k(z) - 1)^2 + 4z^2}}{2z}. \qquad (13)$$

where $R_k(z) = \frac{1-z-\sqrt{\Delta_k(z)}}{4z}$ is the generating function for REs given by grammar (1) but omitting the $\emptyset$, $\Delta_k(z) = 1 - 2z - (7 + 8k)z^2$ and following Nicaud,

$$[z^n]R_k(z) \sim \frac{\sqrt{2(1-\rho_k)}}{8\rho_k\sqrt{\pi}}\rho_k^{-n}n^{-3/2}, \text{ where } \rho_k = \frac{1}{1+\sqrt{8k+8}} \qquad (14)$$

Using the techniques in Broda *et. al* and namely Proposition 3 one has

$$[z^n]R_{\sigma,k}(z) \sim \frac{3}{16\sqrt{\pi}}\left(1 - \frac{b(\rho_k)}{\sqrt{a(\rho_k)}}\right)\sqrt{2(1-\rho_k)}\rho_k^{-(n+1)}n^{-\frac{3}{2}}, \qquad (15)$$

where $a(z)$ and $b(z)$ are polynomials. Thus, the asymptotic ratio of regular expressions with $\sigma \in \mathsf{Pre}(\alpha)$ is:

$$\frac{[z^n]R_{\sigma,k}(z)}{[z^n]R_k(z)} \sim \frac{3}{2}\left(1 - \frac{b(\rho_k)}{\sqrt{a(\rho_k)}}\right). \qquad (16)$$

As $\lim_{k\to\infty} \rho_k = 0$, $\lim_{k\to\infty} a(\rho_k) = 1$, and $\lim_{k\to\infty} b(\rho_k) = 1$, this asymptotic ratio tends to 0 with $k \to \infty$.

Let $i(\alpha)$ be the number of non-disjoint unions appearing during the computation of $\mathsf{Pre}(\alpha)$ originated by the two cases above. Then $i(\alpha)$ verifies

$$\begin{aligned} i(\varepsilon) = i(\sigma) &= 0 & i(\alpha_\sigma^\star\alpha_\sigma) &= i(\alpha_\sigma^\star) + i(\alpha_\sigma) + 1 \\ i(\alpha_\sigma + \alpha_\sigma) &= i(\alpha_\sigma) + i(\alpha_\sigma) + 1 & i(\overline{\alpha_\sigma^\star}\alpha_\sigma) &= i(\overline{\alpha_\sigma^\star}) + i(\alpha_\sigma) \\ i(\alpha_\sigma + \alpha_{\overline{\sigma}}) &= i(\alpha_\sigma) + i(\alpha_{\overline{\sigma}}) & i(\alpha\alpha_{\overline{\sigma}}) &= i(\alpha) + i(\alpha_{\overline{\sigma}}) \\ i(\alpha_{\overline{\sigma}} + \alpha) &= i(\alpha_{\overline{\sigma}}) + i(\alpha) & i(\alpha^\star) &= i(\alpha). \end{aligned}$$

From these equations we can obtain the cost generating function for the number of mergings:

$$I_{\sigma,k}(z) = \frac{(z + z^2)R_{\sigma,k}(z)^2}{\sqrt{\Delta_k(z)}}. \qquad (17)$$

Using again the same Proposition 3 from Broda *et al.*, we conclude that:

$$[z^n]I_{\sigma,k}(z) \sim \frac{1+\rho_k}{64}\frac{\left(a(\rho_k) + b(\rho_k)^2 - 2b(\rho_k)\sqrt{a(\rho_k)}\right)}{\sqrt{\pi}\sqrt{2-2\rho_k}}\rho_k^{-(n+1)}n^{-\frac{1}{2}}. \qquad (18)$$

The cost generating function for the number of letters in $\alpha \in \mathsf{RE}$, computed by Nicaud, is $L_k(z) = \frac{kz}{\sqrt{\Delta_k(z)}}$ and $[z^n]L_k(z) \sim \frac{k\rho_k}{\sqrt{\pi(2-2\rho_k)}}\rho_k^{-n}n^{-1/2}$. With these, we get an asymptotic estimate for the average number of mergings given by:

$$\frac{[z^n]I_{\sigma,k}(z)}{[z^n]L_k(z)} \sim \frac{1-\rho_k}{4\rho_k^2}\lambda_k = \eta_k, \qquad (19)$$

where $\lambda_k = \frac{(1+\rho_k)}{16(1-\rho_k)}\left(a(\rho_k) + b(\rho_k)^2 - 2b(\rho_k)\sqrt{a(\rho_k)}\right)$. It is not difficult to conclude that $\lim_{k\to\infty} \lambda_k = 0$, therefore $\lim_{k\to\infty} \eta_k = 0$. As it is evident from the last two columns of Table 1, for small values of k, the lower bound η_k does not capture all the mergings that occur in $\mathcal{A}_{\mathsf{pre}}$. Although we must study other contributions for those mergings, it seems that for larger values of k, the average number of states of the $\mathcal{A}_{\mathsf{pre}}$ automaton approaches the number of states of the $\mathcal{A}_{\mathsf{pos}}$ automaton.

References

1. Antimirov, V.M.: Partial derivatives of regular expressions and finite automaton constructions. Theor. Comput. Sci. 155(2), 291–319 (1996)
2. Broda, S., Machiavelo, A., Moreira, N., Reis, R.: On the average state complexity of partial derivative automata. Int. J. Found. Comput. Sci. 22(7), 1593–1606 (2011)
3. Broda, S., Machiavelo, A., Moreira, N., Reis, R.: On the average size of Glushkov and partial derivative automata. Int. J. Found. Comput. Sci. 23(5), 969–984 (2012)
4. Champarnaud, J.M., Dubernard, J.P., Jeanne, H., Mignot, L.: Two-sided derivatives for regular expressions and for Hairpin expressions. In: Dediu, A.H., Martín-Vide, C., Truthe, B. (eds.) 7th LATA. LNCS, vol. 7810, pp. 202–213. Springer (2013)
5. Champarnaud, J.M., Ziadi, D.: From Mirkin's prebases to Antimirov's word partial derivatives. Fundam. Inform. 45(3), 195–205 (2001)
6. Champarnaud, J.M., Ziadi, D.: Canonical derivatives, partial derivatives and finite automaton constructions. Theor. Comput. Sci. 289(1), 137–163 (2002)
7. Flajolet, P., Sedgewick, R.: Analytic Combinatorics. CUP (2008)
8. Giammarresi, D., Ponty, J.L., Wood, D.: The Glushkov and Thompson constructions: a synthesis. unpublished manuscript (1998)
9. Glushkov, V.M.: The abstract theory of automata. Russian Mathematical Surveys 16(5), 1–53 (1961)
10. Ilie, L., Yu, S.: Follow automata. Inf. Comput. 186(1), 140–162 (2003)
11. Ko, S., Han, Y.: Left is better than right for reducing nondeterminism of nfas. In: Holzer, M., Kutrib, M. (eds.) 19th CIAA. LNCS, vol. 8587, pp. 238–251. Springer (2014)
12. Maia, E., Moreira, N., Reis, R.: Partial derivative and position bisimilarity automata. In: Holzer, M., Kutrib, M. (eds.) 19th CIAA. LNCS, vol. 8587, pp. 264–277. Springer (2014)
13. Mirkin, B.: An algorithm for constructing a base in a language of regular expressions. Engineering Cybernetics 5, 110–116 (1966)
14. Nicaud, C.: On the average size of Glushkov's automata. In: Dediu, A.H., Ionescu, A.M., Martín-Vide, C. (eds.) 3rd LATA. LNCS, vol. 5457, pp. 626–637. Springer (2009)
15. Thompson, K.: Regular expression search algorithm. Com. ACM 11(6), 410–422 (1968)
16. Yamamoto, H.: A new finite automaton construction for regular expressions. In: Bensch, S., Freund, R., Otto, F. (eds.) NCMA. books@ocg.at, vol. 304, pp. 249–264. Österreichische Computer Gesellschaft (2014)