Information and Computation ••• (••••) •••–•••

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Automata for regular expressions with shuffle ☆

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

CMUP & DCC, Faculdade de Ciências da Universidade do Porto, Rua do Campo Alegre, 4169-007 Porto, Portugal

ARTICLE INFO

Article history:
Received 30 November 2015
Available online xxxx

Keywords:
Regular expressions
Shuffle operation
Partial derivatives
Finite automata
Position automata
Average case
Analytic combinatorics

ABSTRACT

We generalize the partial derivative automaton and the position automaton to regular expressions with shuffle, and study their state complexity in the worst, as well as in the average case. The number of states of the partial derivative automaton ($\mathcal{A}_{pd}$) is, in the worst case, at most 2^m, where m is the number of letters in the expression. The asymptotic average is bounded by $(\frac{4}{3})^m$. We define a position automaton ($\mathcal{A}_{pos}$) that is homogeneous, but in which several states can correspond to a same position, and we show that $\mathcal{A}_{pd}$ is a quotient of $\mathcal{A}_{pos}$. The number of states of the position automaton is at most $1+m(2^m-1)$, while the asymptotic average is no more than $m(\frac{4}{3})^m$.

1. Introduction

The class of regular languages is closed under shuffle (or interleaving operation), and extended regular expressions with shuffle can be much more succinct than the equivalent ones with disjunction, concatenation, and star operators. For the shuffle operation, Mayer and Stockmeyer [16] studied the computational complexity of membership and nonequivalence problems. Nonequivalence is exponential-time-complete, and membership is NP-complete for some classes of regular languages. In particular, they showed that for regular expressions (REs) with shuffle, of size n, an equivalent nondeterministic finite automaton (NFA) needs at most 2^n states, and presented a family of REs with shuffle, of size $\mathcal{O}(n)$, for which the corresponding NFAs have at least 2^n states. Gelade [12], and Gruber and Holzer [14,13] showed that there exists a double exponential trade-off in the translation from REs with shuffle to standard REs. Gelade also gave a tight double exponential upper bound for the translation of REs with shuffle to DFAs. Recently, conversions of shuffle expressions to finite automata were presented by Estrade et al. [9], and Kumar and Verma [15]. In the former an expression is transformed first into a parallel finite automaton and then to an ε-NFA of size 2^{2r-3c}, where r is the size of the expression and c the number of occurrences of the concatenation operator. In the latter the authors give an algorithm for the construction of an ε-free NFA based on the classic Glushkov/position construction, which the authors claim to have at most 2^{m+1} states, where m is the number of letters that occur in the RE with shuffle. Each state corresponds to a set of positions of letters in RE, and in opposition to what happens in the position automaton for standard REs, the automaton is not homogeneous, i.e. the incoming transitions of a state do not share necessarily the same label.

In this paper we present a conversion method of REs with shuffle to ε-free NFAs, by generalizing the partial derivative construction for standard REs [1,17]. For standard REs, the partial derivative automaton ($\mathcal{A}_{pd}$) is a quotient of the Glushkov/position automaton ($\mathcal{A}_{pos}$), and Broda et al. [3,4] showed that, asymptotically, and on average, the size of $\mathcal{A}_{pd}$ is

☆ 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.

* Corresponding author.
E-mail addresses: sbb@dcc.fc.up.pt (S. Broda), ajmachia@fc.up.pt (A. Machiavelo), nam@dcc.fc.up.pt (N. Moreira), rvr@dcc.fc.up.pt (R. Reis).

http://dx.doi.org/10.1016/j.ic.2017.08.013
0890-5401/

half the size of $\mathcal{A}_{pos}$. In the case of REs with shuffle we show that the number of states of the partial derivative automaton is, in the worst case, 2^m (with m as before) and an upper bound for the average size is, asymptotically, $(\frac{4}{3})^m$. We also present a construction of a position automaton $\mathcal{A}_{pos}$ from a RE with shuffle which is homogeneous, and for which the partial derivative automaton is a quotient. The number of states of $\mathcal{A}_{pos}$ is, in the worst case, $1 + m(2^m - 1)$ (with m as before), and an upper bound for the average size is, asymptotically, $m(\frac{4}{3})^m$.

This paper is organized as follows. In the next section we review the shuffle operation and regular expressions with shuffle. In Section 3 we consider equation systems, for languages and expressions, associated with nondeterministic finite automata, and define a solution for a system of equations for a shuffle expression. An alternative and equivalent construction, denoted by $\mathcal{A}_{pd}$, is given in Section 4 using the notion of partial derivative. In Section 5, we give the construction of an automaton based on the notion of positions, denoted by $\mathcal{A}_{pos}$, and show that $\mathcal{A}_{pd}$ is a quotient of $\mathcal{A}_{pos}$. In Section 6, we study the average state complexity of both $\mathcal{A}_{pd}$ and $\mathcal{A}_{pos}$ using the framework of analytic combinatorics. We conclude in Section 7 with some considerations about the upper bounds obtained in this paper, and point out some possible directions for some related future work.

2. Regular expressions with shuffle

Given an alphabet Σ, the shuffle of two words in $\Sigma^\star$ is a finite set of words defined inductively as follows, for $x, y \in \Sigma^\star$ and $a, b \in \Sigma$

$$x \sqcup\!\sqcup \varepsilon = \varepsilon \sqcup\!\sqcup x = \{x\}$$
$$ax \sqcup\!\sqcup by = \{\, az \mid z \in x \sqcup\!\sqcup by \,\} \cup \{\, bz \mid z \in ax \sqcup\!\sqcup y \,\}.$$

This definition is extended to sets of words, i.e., languages, in the natural way:

$$L_1 \sqcup\!\sqcup L_2 = \bigcup_{x \in L_1, y \in L_2} x \sqcup\!\sqcup y.$$

It is well known that if two languages $L_1, L_2 \subseteq \Sigma^\star$ are regular then $L_1 \sqcup\!\sqcup L_2$ is regular. One can extend regular expressions to include the $\sqcup\!\sqcup$ operator. Given an alphabet Σ, we let $\mathsf{T}_{\sqcup\!\sqcup}$ denote the set containing $\emptyset$ plus all terms finitely generated from $\Sigma \cup \{\varepsilon\}$ and operators $+, \cdot, \sqcup\!\sqcup, ^\star$, that is, the expressions τ generated by the grammar

$$\tau \to \emptyset \mid \alpha \tag{1}$$
$$\alpha \to \varepsilon \mid a \mid (\alpha + \alpha) \mid (\alpha \cdot \alpha) \mid (\alpha \sqcup\!\sqcup \alpha) \mid \alpha^\star \quad (a \in \Sigma). \tag{2}$$

As usual, the (regular) language $\mathcal{L}(\tau)$ represented by an expression $\tau \in \mathsf{T}_{\sqcup\!\sqcup}$ is inductively defined as follows: $\mathcal{L}(\emptyset) = \emptyset$, $\mathcal{L}(\varepsilon) = \{\varepsilon\}$, $\mathcal{L}(a) = \{a\}$ for $a \in \Sigma$, $\mathcal{L}(\alpha^\star) = \mathcal{L}(\alpha)^\star$, $\mathcal{L}(\alpha + \beta) = \mathcal{L}(\alpha) \cup \mathcal{L}(\beta)$, $\mathcal{L}(\alpha\beta) = \mathcal{L}(\alpha)\mathcal{L}(\beta)$, and $\mathcal{L}(\alpha \sqcup\!\sqcup \beta) = \mathcal{L}(\alpha) \sqcup\!\sqcup \mathcal{L}(\beta)$. We say that two expressions $\tau_1, \tau_2 \in \mathsf{T}_{\sqcup\!\sqcup}$ are equivalent, and write $\tau_1 \doteq \tau_2$, if $\mathcal{L}(\tau_1) = \mathcal{L}(\tau_2)$. The set of alphabet symbols occurring in an expression τ is denoted by Σ_τ.

Example 1. Consider $\alpha_n = a_1 \sqcup\!\sqcup \cdots \sqcup\!\sqcup a_n$, where $n \geq 1$, $a_i \neq a_j$ for $1 \leq i \neq j \leq n$. Then,

$$\mathcal{L}(\alpha_n) = \{\, a_{i_1} \cdots a_{i_n} \mid i_1, \ldots, i_n \text{ is a permutation of } 1, \ldots, n \}.$$

We recall that standard regular expressions constitute a Kleene algebra and the shuffle operator $\sqcup\!\sqcup$ is commutative, associative, and distributes over $+$. One also has that for all $a, b \in \Sigma$ and $\tau_1, \tau_2 \in \mathsf{T}_{\sqcup\!\sqcup}$,

$$a\tau_1 \sqcup\!\sqcup b\tau_2 \doteq a(\tau_1 \sqcup\!\sqcup b\tau_2) + b(a\tau_1 \sqcup\!\sqcup \tau_2).$$

Given a language L, we define $\varepsilon(\tau) = \varepsilon(\mathcal{L}(\tau))$, where, $\varepsilon(L) = \varepsilon$ if $\varepsilon \in L$ and $\varepsilon(L) = \emptyset$ otherwise. Using the identity elements of $\cdot$ and $+$, and the absorbing property of $\emptyset$, a recursive definition of $\varepsilon : \mathsf{T}_{\sqcup\!\sqcup} \longrightarrow \{\emptyset, \varepsilon\}$ is given by the following: $\varepsilon(a) = \varepsilon(\emptyset) = \emptyset$, $\varepsilon(\varepsilon) = \varepsilon(\alpha^\star) = \varepsilon$, $\varepsilon(\alpha + \beta) = \varepsilon(\alpha) + \varepsilon(\beta)$, $\varepsilon(\alpha\beta) = \varepsilon(\alpha)\varepsilon(\beta)$, and $\varepsilon(\alpha \sqcup\!\sqcup \beta) = \varepsilon(\alpha)\varepsilon(\beta)$. Moreover, in what follows we will always consider expressions reduced according to the following equations $\alpha \sqcup\!\sqcup \varepsilon \doteq \varepsilon \sqcup\!\sqcup \alpha \doteq \alpha$ and $\alpha\varepsilon \doteq \varepsilon\alpha \doteq \alpha$. These are natural simplifications that do not affect the complexity upper bounds obtained.

3. Automata and systems of equations

We first recall the definition of an NFA as a tuple $\mathcal{A} = \langle S, \Sigma, S_0, \delta, F \rangle$, where S is a finite set of states, Σ is a finite alphabet, $S_0 \subseteq S$ the set of initial states, $\delta : S \times \Sigma \longrightarrow \mathcal{P}(S)$ the transition function, and $F \subseteq S$ the set of final states. The extension of δ to sets of states and words is defined by $\delta(X, \varepsilon) = X$ and $\delta(X, ax) = \delta(\cup_{s \in X} \delta(s, a), x)$. A word $x \in \Sigma^\star$ is accepted by $\mathcal{A}$ if and only if $\delta(S_0, x) \cap F \neq \emptyset$. The *language of* $\mathcal{A}$ is the set of words accepted by $\mathcal{A}$ and denoted by $\mathcal{L}(\mathcal{A})$. The *right language of a state* s, denoted by $\mathcal{L}_s$, is the language accepted by $\mathcal{A}$ if we take $S_0 = \{s\}$. If two automata $\mathcal{A}_1$ and $\mathcal{A}_2$ are isomorphic we say that $\mathcal{A}_1 \simeq \mathcal{A}_2$. An equivalence relation $\equiv$ on S is *right invariant* w.r.t. $\mathcal{A}$ if and only if for all $s, t \in S$, $s \equiv t$ implies that

Please cite this article in press as: S. Broda et al., Automata for regular expressions with shuffle, Inf. Comput. (2017), http://dx.doi.org/10.1016/j.ic.2017.08.013