Theoretical Computer Science 627 (2016) 54–70

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Closure properties and descriptional complexity of deterministic regular expressions [☆]

Katja Losemann [*,1], Wim Martens, Matthias Niewerth [1]

Universität Bayreuth, Germany

ARTICLE INFO

Article history:
Received 24 October 2014
Received in revised form 13 February 2016
Accepted 19 February 2016
Available online 26 February 2016
Communicated by Z. Esik

Keywords:
Deterministic regular expressions
One-unambiguous languages
Boolean operations
Automata theory
Descriptional complexity

ABSTRACT

We study the descriptional complexity of regular languages that are definable by deterministic regular expressions, i.e., we examine worst-case blow-ups in size when translating between different representations for such languages. As representations of languages, we consider regular expressions, deterministic regular expressions, and deterministic finite automata. Our results show that exponential blow-ups between these representations cannot be avoided. Furthermore, we study the descriptional complexity of these representations when applying boolean operations. Here, we start by investigating the closure properties of such languages under various language-theoretic operations such as union, intersection, concatenation, Kleene star, and reversal. Our results show that languages that are definable by deterministic regular expressions are not closed under any of these operations. Finally, we show that for all these operations except the Kleene star an exponential blow-up in the size of deterministic regular expressions cannot be avoided.

1. Introduction

Deterministic or *one-unambiguous* regular expressions (henceforth, DREs) have been a topic of research since they were formally defined by Brüggemann-Klein and Wood [2]. Their origins lie in the ISO standard for the Standard Generalized Markup Language (SGML) where they were introduced to ensure efficient parsing. Today, the prevalent schema languages for XML data, such as Document Type Definition (DTD) and XML Schema, require that the regular expressions in their specification are deterministic. From a more foundational point of view, one-unambiguity is a natural manner in which to define determinism in regular expressions. As such, several decision problems behave better for deterministic regular expressions than for general ones. For example, language inclusion for regular expressions is PSPACE-complete but is tractable when the expressions are deterministic. Unfortunately, not every regular language can be expressed by a deterministic expression, i.e., not every regular language is DRE-definable. The canonical example for such a regular language is $L((a+b)^*a(a+b))$, see [2].

Although DRE-definable languages are rather widespread and have been around for quite some time, they are not yet well-understood. This motivates us to study their foundational properties. In particular, we investigate the differences in the descriptional complexity between regular expressions (REs), deterministic regular expressions (DREs), and deterministic finite automata (DFAs). Our initial motivation for this work was an unproved claim in [2] which states that, for expressions

[☆] This article is the extended version of [25].

[*] Corresponding author.
E-mail address: katja.losemann@uni-bayreuth.de (K. Losemann).

[1] Supported by grant number MA 4938/2-1 of the Deutsche Forschungsgemeinschaft (Emmy Noether Nachwuchsgruppe).

http://dx.doi.org/10.1016/j.tcs.2016.02.027
0304-3975/© 2016 Elsevier B.V. All rights reserved.

of the form $\Sigma^* w$, where w is a word over alphabet Σ, every equivalent DRE is at least exponentially larger than the length of w. We proved that this claim is indeed true in the conference version of this work [25], but the proof turned out to be rather non-trivial. The main challenge was that languages of the form $\Sigma^* w$ have polynomial-size REs and DFAs, so one has to develop new techniques for proving lower bounds on the size of DREs. In this article (Section 3), we give two different proofs showing the unavoidable exponential blow-up when translating an RE to a DRE. The first one proves that it cannot be avoided even for finite languages. The latter uses a more general technique which gives more insights in the structure of DRE-definable languages and their DREs.

Another set of contributions in this paper is a study of the effect of language-theoretic operations on languages that are definable by a DRE. In particular, we consider union, intersection, difference, concatenation, Kleene star, and reversal, for unary and arbitrary alphabets. Several of these operations are relevant in XML schema management [9,29]. We provide a complete overview of the closure properties of DRE-definable languages under these operations in Section 4. Afterwards, in Section 5, we briefly investigate the *state complexity* of minimal DFAs for DRE-definable languages. Here, *state complexity* refers to the number of states of the minimal DFA without the sink state. The main reason why we briefly consider state complexity is because we want to provide results that are directly comparable with the results on state complexity in [16,30,33]. That is, the first part of Section 5 lists the increase in state complexity when performing operations on DFAs for DRE-definable languages, if the result of the operation is also DRE-definable. In the second part of Section 5, we study a similar question for DREs. That is, what is the descriptional complexity of DREs that are obtained by performing the aforementioned operations on DREs? Here, we show that for all these operations except the Kleene star an exponential blow-up cannot be avoided when applying the operation on two DREs.

Related work Deterministic regular expressions have recently been investigated from several perspectives [6,12,26,27]. Groz and Maneth proved that the membership problem (is a given word in the language of a given DRE?) can be solved in time $O(m + n \log\log m)$, where n is the size of the word and m the size of the expression [12]. The *DRE-definability* problem asks whether a given regular expression or non-deterministic automaton defines a language that can be expressed by a DRE and was recently proved to be PSPACE-complete [6,26].

Deterministic regular expressions *with counters* are also a topic of investigation [5,11,17,20,21], since these expressions are the ones used to define content models in XML Schema. In fact, determinism for regular expressions with counters can be defined in different ways (weak determinism and strong determinism) [11]. While the expressiveness of strongly deterministic expressions with counting is the same as DREs, the weakly deterministic expressions, which are the ones used in XML Schema, are more expressive [11]. However, weakly deterministic regular expressions with counting still cannot define all regular languages [11]. It was recently shown that it can be decided if the language of a given finite automaton is expressible by a weakly deterministic regular expression with counting [23].

In this article we focus on *descriptional complexity* of DREs. Research on descriptional complexity of regular languages focused mainly on REs and DFAs. It is well-known that an exponential blow-up cannot be avoided when translating an RE into a DFA [16]. Ehrenfeucht and Zeiger [7] proved that there also exist DFAs which are exponentially more succinct than each equivalent RE. Gruber and Holzer [13,15] showed that there exist certain characteristics of automata which make equivalent regular expressions large. However, these characteristics cannot naïvely be transferred to DREs. For example, the languages used in the literature for proving lower bounds on the size of REs (e.g., [7,13,15]) are not definable by DREs.

The state complexity of boolean operations on DFAs is studied in [22,28,30,32,33], where in [30] the focus is on unary languages. In Section 5.1 we see that many results in [33] directly apply for DRE-definable languages since they concern finite languages and every finite language is DRE-definable [1].

Gelade and Neven [10] and Gruber and Holzer [14] independently examined the descriptional complexity of complementation and intersection for REs. They showed that the size of the smallest RE for the intersection of a fixed number of REs can be exponential; and that the size of the smallest RE for the complement of an RE can be double-exponential. Furthermore, these bounds are tight. Gelade and Neven also investigate these operations on DREs and proved that the exponential bound on intersection is also tight when the input is given as DREs instead of REs [10]. Moreover, they proved that the complement of a DRE can always be described by a polynomial-size RE. However, in their proofs, the languages of the resulting REs are not DRE-definable. Concatenation and reversal operations on regular languages are studied in [3,18,19,31,34], where in [34] also languages over unary alphabets are examined.

2. Definitions

By Σ we always denote a finite alphabet of symbols. A $(\Sigma\text{-})$*word* w over alphabet Σ is a finite sequence of symbols $a_1 \cdots a_n$, where $a_i \in \Sigma$ for each $i = 1, \ldots, n$. The set of all Σ-words is denoted by Σ^*. The *length* of a word $w = a_1 \cdots a_n$ is n and is denoted by $|w|$. The empty word is denoted by ε. A *(word) language* L is a set of words. For two languages L_1 and L_2, we define the concatenation $L_1 \cdot L_2$ as the set $\{vw \mid v \in L_1 \wedge w \in L_2\}$. By L^i with $i \in \mathbb{N}$ we denote the concatenation $L \cdots L$ of i-times the language L.

A *(deterministic, finite) automaton* (or *DFA*) A is a tuple $(Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, the transition function $\delta : Q \times \Sigma \nrightarrow Q$ is a partial function, q_0 is the initial state, and $F \subseteq Q$ is the set of accepting states. We say that the aforementioned transition is q_1*-outgoing*, q_2*-incoming*, or a*-labeled*. The *run of* A *on word* $w = a_1 \cdots a_n$ is a sequence $q_0 \cdots q_n$ where, for each $i = 1, \ldots, n$, $\delta(q_{i-1}, a_i) = q_i$. The word w is *accepted* by A if the run is *accepting*, i.e., if $q_n \in F$. By $L(A)$ we