Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Concatenation-free languages

Martin Kutrib *, Matthias Wendlandt

Institut für Informatik, Universität Giessen, Arndtstr. 2, 35392 Giessen, Germany

A R T I C L E I N F O

Article history:
Received 30 November 2015
Received in revised form 2 August 2016
Accepted 18 August 2016
Available online xxxx

Keywords:
Concatenation-free languages
Regular expressions
Expressive capacity
Characterizations
Closure properties
Subregular hierarchy

A B S T R A C T

The expressive capacity of three different types of regular expressions without concatenation is studied. In particular, we consider alphabetic concatenation-free expressions, which are ordinary regular expressions without concatenation, simple concatenation-free expressions, where the set of literals is a finite set of words instead of letters, and concatenation-free expressions, where additionally complementation operations are possible. Characterizations of the corresponding language classes are obtained. In particular, a characterization of unary concatenation-free languages by the Boolean closure of certain sets of languages is shown. The characterizations are then used to derive a strict hierarchy that is, in turn, strictly included in the family of regular languages. The closure properties of the families are investigated. Furthermore, the position of the family of concatenation-free languages in the subregular hierarchy is considered and settled for the unary case. In particular, there are concatenation-free languages that do not belong to any of the families in the hierarchy. Moreover, except for comets, all the families considered in the subregular hierarchy are strictly included in the family of concatenation-free languages.

1. Introduction

The investigation of regular expressions originates in [10]. They allow a set-theoretic characterization of languages accepted by finite automata. Compared to automata, regular expressions may be better suited for human users and therefore are often used as interfaces to specify certain patterns or languages. For example, regular(-like) expressions can be found in many software tools, where the syntax used to represent them may vary, but the concepts are very much the same everywhere. The leading idea is to describe languages by using constants and operator symbols. Classically, the constants are literals from the underlying alphabet and the symbol for the empty set, and operations are union, concatenation, and Kleene star. However, the regular languages are closed under many more operations. So, adding these operations to regular expressions cannot increase their expressive power. On the other hand, removing an operation or replacing it by another may decrease the expressive capacity. For example, replacing the star by complementation yields the well-known and important subregular family of star-free (or regular non-counting) languages [4]. This family obeys nice characterizations, for example, in terms of aperiodic syntactic monoids [17], permutation-free DFA [13], and loop-free alternating finite automata [16]. See [6,8,11] for surveys of the complexity of regular(-like) expressions.

Here we study the expressive power of three different types of regular expressions without concatenation. In analogy with the definition of union-free languages [14] we consider *alphabetic concatenation-free expressions* where the operation of concatenation is not available. Then, in order to allow non-trivial languages to be expressed, we allow any finite set of

* Corresponding author.
E-mail addresses: kutrib@informatik.uni-giessen.de (M. Kutrib), matthias.wendlandt@informatik.uni-giessen.de (M. Wendlandt).

http://dx.doi.org/10.1016/j.tcs.2016.08.014

words as literals, in this way defining *simple concatenation-free expressions*. Finally, in analogy with the star-free expressions we study *concatenation-free expressions*, where the concatenation is traded for complementation.

The paper is organized as follows. In the next section, we present the basic notations and definitions, and provide an introductory example. Section 3 is devoted to explore the limits of the expressive capacity of the expressions. The basic questions are whether the different definitions lead to a hierarchy of language families and whether the strongest class captures the regular languages or not. The first question is answered in the affirmative while the latter question is answered negatively. To this end, characterizations of the three language classes are given. The characterizations are then used to derive witnesses for the separations. Furthermore, concatenation hierarchies are shown for two classes, while for the third class it is shown that just one concatenation operation suffices to describe all *unary* regular languages. The closure properties of the families under consideration are summarized and complemented in Section 4. The properties are represented in Table 1. Finally, in Section 5 the position of the family of concatenation-free languages in the hierarchy of several subregular language families (see Fig. 2) is considered. It turns out that there are concatenation-free languages that do not belong to any other of the subregular families. For the special case of unary languages the precise position can be settled. In detail, though unary concatenation-free expressions are not as expressive as general regular expressions, they yield a language family that strictly includes all the families of the subregular hierarchy depicted in Fig. 2, except for the (two-sided) comets to which it is incomparable.

2. Preliminaries and definitions

We write Σ^* for the set of all words over the finite alphabet Σ. The *empty word* is denoted by λ. For the *length* of w we write $|w|$. We use $\subseteq$ for *inclusions* and $\subset$ for *strict inclusions*. The *complement* of a language L over alphabet Σ is again a language over alphabet Σ which is denoted by $\overline{L}$. The *family of finite languages* is denoted by FIN.

The *regular expressions* over an alphabet Σ and the languages they describe are defined inductively in the usual way: $\emptyset$ and every word (of length one) $v \in \Sigma$ are regular expressions, and when s and t are regular expressions, then $(s \cup t)$, $(s \cdot t)$, and $(s)^*$ are also regular expressions. The language $L(r)$ defined by a regular expression r is defined as follows: $L(\emptyset) = \emptyset$, $L(v) = \{v\}$, $L(s \cup t) = L(s) \cup L(t)$, $L(s \cdot t) = L(s) \cdot L(t)$, and $L(s^*) = L(s)^*$.

Since the regular languages are closed under many more operations, the approach to add operations like intersection ($\cap$), complementation ($\overline{}$), or squaring (^{2}) does not increase the expressive power of regular expressions. However, replacing operations by others may decrease the expressive power. So, in general, $\text{RE}(\Sigma, \Lambda, \Phi)$, where $\Lambda \subset \Sigma^*$ is a finite set of initial words, and Φ is a set of (regularity preserving) operations, denotes all regular(-like) expressions over Λ using only operations from Φ. Hence $\text{RE}(\Sigma, \Sigma, \{\cup, \cdot, *\})$ refers to the set of all ordinary regular expressions, and $\text{RE}(\Sigma, \Sigma, \{\cup, \cdot, \overline{}\})$ defines the star-free languages.

Here we study the expressive power of three different types of regular expressions without concatenation. In particular, we complement the study of union-free languages [14], where a language is said to be *union free* if it can be described by a regular expression that does not contain the union operation. Here, we consider *alphabetic concatenation-free languages* where the operation of concatenation is not available, that is, $\text{RE}(\Sigma, \Sigma, \{\cup, *\})$. Since in the presence of concatenation, every word in Λ can be obtained by concatenating letters from Σ, the set Λ can be created for free. Here, however, we do not have concatenation and, thus, consider to provide initially a *finite set* of words. The corresponding expressions $\text{RE}(\Sigma, \Lambda, \{\cup, *\})$ define the *simple concatenation-free languages*. Finally, we adapt the definition of star-free languages and trade the concatenation for complementation, that is, we investigate regular expressions from $\text{RE}(\Sigma, \Lambda, \{\cup, *, \overline{}\})$ and call the family of languages represented by such expressions *concatenation-free languages*.

Trivially, if even alphabetic concatenation-free expressions are extended such that an arbitrary number of concatenations is allowed, all regular languages are captured. A natural question is whether the number of concatenations necessary to express any regular language is bounded. So, for any fixed integer $k \geq 0$, the set of (extended) concatenation-free expressions that may contain at most k applications of the operation concatenation is referred to as *concatenation-free expressions of degree k*. The family of languages represented by such expressions is referred to by *concatenation-free languages of degree k*.

For convenience, parentheses in regular expressions are sometimes omitted, where it is understood that the unary operations complementation and star have a higher priority than union.

In order to clarify our notion, we continue with an example.

Example 1. Let $L \subseteq \{a, b\}^*$ be the language of words that either begin with an a and have at least two consecutive b, or begin with a b.

Language L is described by the concatenation-free expression $r = \overline{(a \cup ab)^*}$. The subexpression $(a \cup ab)^*$ gives all words over the alphabet $\{a, b\}$ beginning with an a that do not have the factor bb. The complement of r describes L. □

3. Limits of concatenation-free expressions

In this section we study the expressive power of the three different types of concatenation-free expressions and compare them with each other.