

EXPRESSIVE CAPACITY OF SUBREGULAR EXPRESSIONS[☆]

MARTIN KUTRIB^{*} AND MATTHIAS WENDLANDT

Abstract. Different types of subregular expressions are studied. Each type is obtained by either omitting one of the regular operations or replacing it by complementation or intersection. For uniformity and in order to allow non-trivial languages to be expressed, the set of literals is a finite set of words instead of letters. The power and limitations as well as relations with each other are considered, which is often done in terms of unary languages. Characterizations of some of the language families are obtained. A finite hierarchy is shown that reveals that the operation complementation is generally stronger than intersection. Furthermore, we investigate the closures of language families described by regular expressions with omitted operation under that operation. While it is known that in case of union this closure captures all regular languages, for the cases of concatenation and star incomparability results are obtained with the corresponding language families where the operation is replaced by complementation.

Mathematics Subject Classification. 68Q45, 68Q15.

Accepted November 21, 2018.

1. INTRODUCTION

The investigation of regular expressions originates in [6]. 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. It is well known that the operations union, concatenation, and star yield expressions that capture the regular languages. Due to the strong closure properties of regular languages, several more operations could be added to the expressions without increasing their expressive capacity. On the other hand, removing some operation or replacing it by another one may have an impact on the expressive power. For example, replacing the star by complementation yields the well-known and important subregular family of star-free (or regular non-counting) languages [1] that obeys nice characterizations, for example, in terms of aperiodic syntactic monoids [16], permutation-free DFA [10], and loop-free alternating finite automata [15]. Recently, the concatenation-free languages have been studied, where the concatenation is replaced by complementation [7].

[☆]A preliminary version of this paper appeared in the Proceeding of the Workshop on Non-Classical Models of Automata and Applications (NCMA 2016) [8].

Keywords and phrases: Regular expressions, concatenation-free languages, star-free languages, union-free languages, expressive capacity, characterizations, closure properties, subregular hierarchy.

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

^{*} Corresponding author: kutrib@informatik.uni-giessen.de

Playing around with different sets of operations allowed, many important results on the complexity of decision problems have been obtained, for example, in [2, 5, 13, 14, 18, 19]. A recent survey on the complexity of regular(-like) expressions can be found in [3].

Here, we study different types of subregular expressions. Each type is obtained by either omitting one of the classical operations or replacing it by complementation or intersection. Furthermore, we investigate the closures of language families described by regular expressions with omitted operation under that operation. From a structural perspective this means that the omitted operation is now allowed but at the outer level of the expressions only. Our main interest is on the expressive power and limitations of these subregular expressions as well as on their mutual relations. However, in order to allow non-trivial languages to be expressed, we allow any finite set of words as literals.

The paper is organized as follows. In the next section, we present the basic notations and definitions, and provide an introductory example. Moreover, in view of the fact that the operation star plays often a crucial role for generating an infinite language out of a finite one, we present a preliminary result on the star of unary regular languages that is often used in the sequel. Section 3 is devoted to explore the limits of the expressive capacity of union-free expressions, that is, regular expressions where the union is either traded for complementation or intersection (the term union free has been used differently before in [12] for ordinary regular expressions where the operation of union is removed). An immediate question is whether or not all regular languages can be described by such expressions, and if not, how they are related. We are going to answer the first question negatively for both expression classes and show that the unary languages described by the latter class are properly included in the languages described by the former class. To this end, it is shown that the unary languages are all of a certain form obtained by stretching a cofinite language and extending the lengths of the words by a finite set of numbers. In Section 4 expressions are considered, where the concatenation is replaced by the intersection. The case where the concatenation is traded for complementation has recently been studied in [7]. A characterization of the unary languages in question by the union closure of certain languages shows a strict inclusion of the latter family in the former one. Some further results for non-unary languages are presented as well. Section 5 explores the relations with the well-known family of star-free languages as well as with the family of languages described by regular expressions where the star is traded for intersection. It turns out that the latter family has a quite simple characterization as they coincide with the finite languages. Finally, Section 6 first complements the picture of what is known about subregular expressions where one operation is omitted. Then it deals with the closures of language families described by regular expressions with omitted operation under that operation. While it is known that in case of union this closure captures all regular languages, for the cases of concatenation and star incomparability results are obtained with the corresponding language families where the operation is replaced by complementation. The hierarchical structure summarizing main results is presented in Figure 3.

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 \bar{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) or complementation ($\bar{}$) 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 literal words, and Φ is a set of (regularity preserving) operations, denotes all regular expressions

over finite subsets of Λ using only operations from Φ . Hence $\text{RE}(\Sigma, \Sigma, \{\cup, \cdot, *\})$ or REG refers to the set of all ordinary regular expressions, and $\text{RE}(\Sigma, \Sigma, \{\cup, \cdot, \bar{}\})$ defines the star-free languages [1].

Here, we study the expressive power of different types of regular expressions where one of the three ordinary regular operations is omitted. Moreover, the idea of the definition of star-free languages is extended, that is, to trade the star for complementation, to all of the three ordinary regular operations. Furthermore, we systematically study the expressive power of regular expressions where each of the ordinary operations is replaced by the intersection. 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. Moreover, in the presence of union, every finite subset of words in Λ can be created for free. Here, however, we do not have necessarily concatenation or union and, thus, provide initially *finite subsets* of words as literals in order to allow non-trivial languages to be expressed. Moreover, we do this for uniformity and comparability for all types in question. The corresponding expressions $\text{RE}(\Sigma, \Lambda, \{\cup, *\})$ define the *simple concatenation-free languages*. Similarly, the expressions $\text{RE}(\Sigma, \Lambda, \{\cup, \cdot\})$ define the *simple star-free languages*, and the expressions $\text{RE}(\Sigma, \Lambda, \{\cdot, *\})$ define the *simple union-free languages*. In accordance with star-free expressions, we call $\text{RE}(\Sigma, \Lambda, \{\cup, *, \bar{}\})$ *concatenation-free* expressions that have been studied in [7] before, and denote expressions from $\text{RE}(\Sigma, \Lambda, \{\cdot, *, \bar{}\})$ *union free*. Here, it has to be mentioned that the term union free has been used differently in [12] for ordinary regular expressions where the operation of union is not available. The remaining three types $\text{RE}(\Sigma, \Lambda, \{\cup, *, \cap\})$, $\text{RE}(\Sigma, \Lambda, \{\cup, \cdot, \cap\})$, and $\text{RE}(\Sigma, \Lambda, \{\cdot, *, \cap\})$ are referred to as *intersection–concatenation-free*, *intersection–star-free*, and *intersection–union-free* expressions. We use the same notations for the families of languages described.

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, intersection, and concatenation, and that the concatenation has a higher priority than union and intersection.

We are also interested in closures of language families described by simple regular expressions, where the closure is built under the operation omitted. In general, let \mathcal{L} be a family of languages and op be one of the operations union (\cup), concatenation (\cdot), or star ($*$). Then $\Gamma_{op}(\mathcal{L})$ denotes the *least family of languages which contains all members of \mathcal{L} and is closed under op* .

Regular expressions and, thus, the languages described can be represented by *expression trees*. The leaves of such trees are labeled with finite subsets of words from Λ , the literals of the expression. The inner nodes are labeled by operations from Φ , where an inner node labeled by a unary operation has exactly one successor and an inner node labeled by a binary operation has exactly two successors. An expression tree is evaluated from bottom to top by attaching languages to the nodes. The leaves are attached with their labels. The language attached to an inner node is the result of applying the operation of its label to the language(s) attached to its successors. So, the language described by the regular expression is attached to the root of its expression tree (see Fig. 1).

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

Example 2.1. The unary language $L = \{a\} \cup \{a^{5^n} \mid n \geq 0\}$ is described by the union-free expression $r = \{aa\} \cdot \overline{\{aaa\} \cdot \overline{\{aaaaa\}}^*}$. The expression tree of r is depicted in Figure 1.

The expression $\{aaaaa\}^*$ describes all words whose lengths are congruent 0 modulo 5, that is, $L_4 = \{a^n \mid n \equiv 0 \pmod{5}\}$. The concatenation of the word aaa results in all words whose lengths are congruent 3 modulo 5, that is, $L_3 = \{a^n \mid n \equiv 3 \pmod{5}\}$. Then the complementation gives language L_2 which is $\{a^n \mid n \not\equiv 3 \pmod{5}\}$. The following concatenation with the word aa describes the language L_1 of all words whose lengths are at least 2 and are not congruent 0 modulo 5, that is, $L_1 = \{a^n \mid n \geq 2 \text{ and } n \not\equiv 0 \pmod{5}\}$. Finally, the complement of L_1 is language L . ■

Next, we turn to unary languages obtained by applying the star operation to a finite set of words. To this end, we recall a well-known useful fact which is related to number theory and Frobenius numbers (see, for example, [17] for a survey).

Lemma 2.2. *Let x_1, x_2, \dots, x_k be positive integers. Then every sufficiently large integer can be written as a non-negative integer linear combination of the x_i if and only if their greatest common divisor $\text{gcd}(x_1, x_2, \dots, x_k)$*

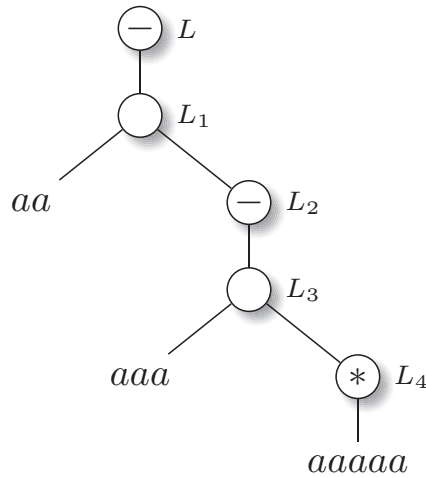


FIGURE 1. The union-free expression $r = \overline{\{aa\} \cdot \overline{\{aaa\} \cdot \overline{\{aaaaa\}^*}}$ represented in a tree. The languages attached to the nodes are explained in Example 2.1.

equals 1, that is, if they are relatively prime. The largest positive integer which cannot be represented as a non-negative integer linear combination of the x_i is their Frobenius number $g(x_1, x_2, \dots, x_k)$. In particular, for $k = 2$, the largest integer that cannot be represented is $x_1x_2 - x_1 - x_2$.

Let $m \geq 1$ be an integer. A unary language $L \subseteq \{a\}^*$ is stretched by m to a language $L_{(m)}$ in the following sense: $L_{(m)} = \{a^{m \cdot n} \mid a^n \in L\}$. That is, a unary language is stretched by multiplying each word length by m . We say that language $L_{(m)}$ is obtained from L stretched by m . Combining this definition and the previous lemma yields the next result which is exploited in the sequel. A similar result has been obtained in [7].

Lemma 2.3. *Let $L \subseteq \{a\}^*$ be an arbitrary unary language that includes at least one non-empty word. Then there exists $m \geq 1$ such that L^* is obtained from a cofinite language that is stretched by m .*

Proof. First, assume that $L = \{a^m\}$, for some $m \geq 1$, is a singleton. Then we have $L^* = \{a^{i \cdot m} \mid i \geq 0\}$, and L^* is obtained from $\{a\}^*$ stretched by m . Second, if L contains two words whose lengths p and q are relatively prime, then L^* is a superset of $\{a^n \mid n > pq - p - q\}$ by Lemma 2.2. So, L^* is obtained from a cofinite language stretched by 1.

For the last case we assume that the lengths of each two different words in L are *not* relatively prime. Moreover, we denote the greatest common divisor of the lengths of *all* words in L by m , where it may happen that $m = 1$. Now we consider the language of words of L whose lengths are divided by m : $L' = \{a^n \mid a^{m \cdot n} \in L\}$. We choose an arbitrary word $a^{n_0} \in L'$ and denote the prime factors of n_0 by d_1, d_2, \dots, d_k . Next we choose some word $a^{n_1} \in L'$ whose length does not contain the prime factor d_1 . Such a word must exist since otherwise all word lengths in L' would be divisible by d_1 which is a contradiction to the definition of L' . Similarly, we continue to choose some (not necessarily distinct) words $a^{n_i} \in L'$, $2 \leq i \leq k$, whose lengths are not divisible by d_i , respectively. We conclude that $\gcd(n_0, n_1, \dots, n_k) = 1$. So, again by Lemma 2.2 it follows that there exists a positive integer ℓ , that is the Frobenius number $g(n_0, n_1, \dots, n_k)$, such that $L'^* \supseteq \{a^n \mid n > \ell\}$. This implies L'^* is cofinite and, therefore, L^* is obtained from the cofinite language L'^* stretched by m . \square

3. EXPRESSIVE CAPACITY OF UNION-FREE EXPRESSIONS

We start to investigate the expressive limits of intersection–union-free and union-free expressions. An immediate question is whether or not all regular languages can be described by such expressions, and if not, how they are related. We are going to answer the first question negatively for both expression classes.

A unary language is said to be of form $\langle m, \Delta \rangle$ if it is obtained from a cofinite language whose words are stretched by $m \geq 1$ and then extended by a finite set of lengths $\Delta = \{x_1, x_2, \dots, x_k\}$, $k \geq 1$, $x_i \geq 0$, $1 \leq i \leq k$. That is, if L is obtained from a cofinite language stretched by m , then language

$$\bigcup_{w \in L} \{a^{|w|+x_1}, a^{|w|+x_2}, \dots, a^{|w|+x_k}\}$$

is of form $\langle m, \Delta \rangle$. In particular, this means that every cofinite unary language is of form $\langle 1, \{0\} \rangle$ and vice versa. If a language L is of form $\langle m, \Delta \rangle$, then we say that $L \cup \{\lambda\}$ is of form $\langle m, \Delta \rangle \cup \{\lambda\}$. Moreover, we say that a language L is of form $\langle m, \Delta \rangle \cup \{\lambda\}$ with finite error set E , if E is a finite set of positive integers, there exists a language L' that is of form $\langle m, \Delta \rangle \cup \{\lambda\}$, and $L' = L \cup \{a^x \mid x \in E\}$. So, a finite set of words whose lengths are at least one is added to L in order to obtain a language of form $\langle m, \Delta \rangle \cup \{\lambda\}$.

In order to derive a unary language that is not intersection–union free, later we will argue along expression trees. During the evaluation, the three situations dealt with in the next three lemmas are of particular interest.

Lemma 3.1. *Let $L \subseteq \{a\}^*$ be a unary language of form $\langle m, \Delta \rangle \cup \{\lambda\}$ with finite error set E , $0 \in \Delta$, and the property that $a \in L$ implies $1 \in \Delta$. Then the concatenation of L and a finite language L_{fin} including the empty word is of form $\langle m, \Delta' \rangle \cup \{\lambda\}$ with finite error set E' and $0 \in \Delta'$. Moreover, if a belongs to the concatenation, then $1 \in \Delta'$.*

Proof. Let $\Delta = \{0, x_2, x_3, \dots, x_k\}$ and $L_{\text{fin}} = \{\lambda, a^{y_2}, a^{y_3}, \dots, a^{y_l}\}$. Then the concatenation by L_{fin} extends every word in L by the lengths $\{0, y_2, y_3, \dots, y_l\}$. That is, $L \cdot L_{\text{fin}}$ is of form $\langle m, \Delta' \rangle \cup \{\lambda\}$ with finite error set E' , where

$$\Delta' = \{0, y_2, y_3, \dots, y_l, x_2, x_2 + y_2, x_2 + y_3, \dots, x_2 + y_l, \dots, x_k, x_k + y_2, \dots, x_k + y_l\}.$$

The word a belongs to the concatenation if and only if it belongs to L or to L_{fin} . In the former case we have $1 \in \Delta$ by assumption and, thus, $1 \in \Delta'$. In the latter case $1 \in \{y_2, y_3, \dots, y_l\}$. So, in both cases we obtain $1 \in \Delta'$. \square

Next we consider concatenations of infinite languages with certain properties.

Lemma 3.2. *Let $L_1 \subseteq \{a\}^*$ be a unary language of form $\langle m_1, \Delta_1 \rangle \cup \{\lambda\}$ with finite error set E_1 , $0 \in \Delta_1$, and the property that $a \in L_1$ implies $1 \in \Delta_1$, and $L_2 \subseteq \{a\}^*$ be a unary language of form $\langle m_2, \Delta_2 \rangle \cup \{\lambda\}$ with finite error set E_2 , $0 \in \Delta_2$, and the property that $a \in L_2$ implies $1 \in \Delta_2$. Then, for some number m_3 and set Δ_3 , the concatenation of L_1 and L_2 is of form $\langle m_3, \Delta_3 \rangle \cup \{\lambda\}$ with finite error set E_3 and $0 \in \Delta_3$. Moreover, if a belongs to the concatenation, then $1 \in \Delta_3$.*

Proof. Let $L_1 \cup \{a^x \mid x \in E_1\}$ be obtained from the cofinite language L'_1 whose words are stretched by $m_1 \geq 1$ and then extended by the lengths of Δ_1 , and accordingly L_2 be obtained from the cofinite language L'_2 . We consider the lengths of the longest word not belonging to L'_1 plus one and of the longest word in L'_1 that may be added due to the error set, and denote the length of the longer of both by c_1 . Similarly, c_2 is used for the length of the corresponding word for L'_2 . In this way all words added by the error sets are excluded for a moment.

The concatenation of L_1 and L_2 includes all words whose lengths have a representation as $(c_1 + s_1)m_1 + x + (c_2 + s_2)m_2 + y$, where $s_1, s_2 \geq 0$, $x \in \Delta_1$, and $y \in \Delta_2$. These are all words whose lengths have a representation as $s_1m_1 + s_2m_2 + c_1m_1 + c_2m_2 + x + y$. Now, we derive from Lemma 2.2 that the set $\{s_1m_1 + s_2m_2 \mid s_1 \geq 0, s_2 \geq 0\}$ is obtained from a cofinite set stretched by $m_3 = \text{gcd}(m_1, m_2)$. Since

$$c_1m_1 + c_2m_2 + m_1m_2 = (c_1 + m_2)m_1 + c_2m_2 = c_1m_1 + (c_2 + m_1)m_2,$$

we conclude that also the set $\{(c_1 + s_1)m_1 + (c_2 + s_2)m_2 \mid s_1 \geq 0, s_2 \geq 0\}$ is obtained from a cofinite set stretched by m_3 . That is, it is of form $\langle m_3, \{0\} \rangle$.

Finally, the language of words whose lengths have a representation as

$$(c_1 + s_1)m_1 + x + (c_2 + s_2)m_2 + y$$

are of form $\langle m_3, \Delta_3 \rangle$ with $\Delta_3 = \{x + y \mid x \in \Delta_1 \text{ and } y \in \Delta_2\}$. In particular, we have $0 \in \Delta_3$ since $0 \in \Delta_1 \cap \Delta_2$. If a belongs to the concatenation, then it belongs to L_1 or to L_2 . By assumption, in the former case $1 \in \Delta_1$ and in the latter case $1 \in \Delta_2$. Again, since $0 \in \Delta_1 \cap \Delta_2$ we derive $1 \in \Delta_3$.

All words in the concatenation of L_1 and L_2 not considered so far, have a representation as $d_1m_1 + x + d_2m_2 + y$, where $d_1 \geq 0$ and $d_2 \geq 0$ are constants. Therefore, their lengths are multiples of m_3 extended by $x + y$ as well. Since all but a finite number of them have been shown to belong to the concatenation of L_1 and L_2 , the possible error set E_3 to make the concatenation of form $\langle m_3, \Delta_3 \rangle$ is finite. Finally, since the empty word belongs to both languages, we conclude that the concatenation L_1L_2 is of form $\langle m_3, \Delta_3 \rangle \cup \{\lambda\}$ with finite error set E_3 , $0 \in \Delta_3$, and a in the concatenation implies $1 \in \Delta_3$. \square

The next lemma deals with the intersection.

Lemma 3.3. *Let $L_1 \subseteq \{a\}^*$ be a unary language of form $\langle m_1, \Delta_1 \rangle \cup \{\lambda\}$ with finite error set E_1 , $0 \in \Delta_1$, and the property that $a \in L_1$ implies $1 \in \Delta_1$, and $L_2 \subseteq \{a\}^*$ be a unary language of form $\langle m_2, \Delta_2 \rangle \cup \{\lambda\}$ with finite error set E_2 , $0 \in \Delta_2$ and the property that $a \in L_2$ implies $1 \in \Delta_2$. Then the intersection of L_1 and L_2 is of form $\langle m_3, \Delta_3 \rangle \cup \{\lambda\}$ with finite error set E_3 and $0 \in \Delta_3$. Moreover, if a belongs to the intersection, then $1 \in \Delta_3$.*

Proof. If at least one of m_1 and m_2 is equal to 1 we argue as follows. If $m_1 = 1$ we know that L_1 is cofinite. So, the intersection of L_1 and L_2 includes all but finitely many words from L_2 including the empty word. If a belongs to the intersection it belongs to L_2 and, thus, $1 \in \Delta_2$. Now only a finite set of words has to be added to make the intersection of form $\langle m_2, \Delta_2 \rangle \cup \{\lambda\}$ with finite error set E_3 . The case $m_2 = 1$ is treated analogously.

If $m_1 \geq 2$ and $m_2 \geq 2$, the lengths of all words in the intersection of L_1 and L_2 satisfy the equation $s_1m_1 + x = s_2m_2 + y$, for some $s_1, s_2 \geq 0$, $x \in \Delta_1$, and $y \in \Delta_2$. The following consideration is done for all pairs $(x, y) \in \Delta_1 \times \Delta_2$ separately.

We distinguish two cases. First, assume that m_1 and m_2 are relatively prime. If $x = y$ then $s_1 = s_2 = 0$ solves the equation above. If $x \neq y$, say $x > y$, we consider the equation $s_1m_1 + (x - y) = s_2m_2$. Since m_1 and m_2 are relatively prime, there exists a $j_0 \geq 1$ such that $j_0m_2 \equiv 1 \pmod{m_1}$. So, there exists an $i_0 \geq 0$ such that $j_0m_2 = i_0m_1 + 1$. We conclude $(x - y)j_0m_2 = (x - y)i_0m_1 + (x - y)$ and, thus, the equation has a solution. The argumentation for $y > x$ is analogously. Moreover, for $x = y$ as well as for $x \neq y$, to find larger solutions one has to add numbers that are multiples of m_1 as well as of m_2 . On the other hand, adding m_1m_2 to a solution gives the next one. Let $z_{x,y}$ denote the smallest solution of the equation such that $a^{z_{x,y}}$ belongs to the intersection of L_1 and L_2 . Then, for fixed x and y , all words in the intersection have a representation as $s_3m_1m_2 + z_{x,y}$, for $s_3 \geq 0$.

In order to complete the proof that the intersection of L_1 and L_2 is of form $\langle m_1m_2, \Delta_3 \rangle$ with finite error set and $\Delta_3 = \{z_{x,y} \mid (x, y) \in \Delta_1 \times \Delta_2\}$ it remains to be shown that whenever $im_1m_2 + z_{x,y}$, for some fixed $i \geq 0$, is the length of a word belonging to the intersection, then the words with lengths $im_1m_2 + z_{x',y'}$ belong to the intersection as well. Recall that $z_{x,y} = s_1m_1 + x = s_2m_2 + y$, for some fixed s_1 and s_2 . So, $im_1m_2 + s_1m_1 + x$ is the length of a word in L_1 . This implies that $im_1m_2 + s_1m_1 + x'$ is the length of a word in L_1 as well. Similarly, we obtain that $im_1m_2 + s_2m_2 + y'$ is the length of a word in L_2 . Since $L_1 \cup \{a^x \mid x \in E_1\}$ is obtained from the cofinite language whose words are stretched by $m_1 \geq 1$ and then extended by the lengths of Δ_1 , for all but a finite number of words in L_1 , adding another m_1 symbols a yields again a word in L_1 . The same is true for L_2 and m_2 . Now let $z_{x',y'}$ be $s'_1m_1 + x' = s'_2m_2 + y'$. Then in all but a finite number of cases $im_1m_2 + (s'_1 - s_1 + s_1)m_1 + x'$ is the length of a word in L_1 and in all but a finite number of cases $im_1m_2 + (s'_2 - s_2 + s_2)m_2 + y'$ is the length of a word in L_2 . Since $s'_1m_1 + x' = s'_2m_2 + y'$, the word with length $im_1m_2 + z_{x',y'}$ belongs to the intersection in all but a finite number of cases. Now only a finite set of words has to be added to make the intersection of form $\langle m_1m_2, \Delta_3 \rangle$ with finite error set and $\Delta_3 = \{z_{x,y} \mid (x, y) \in \Delta_1 \times \Delta_2\}$.

Next we turn to the second case where m_1 and m_2 are not relatively prime. Let $\gcd(m_1, m_2) = d$ for some $d > 1$ and $m_1 = dm'_1$ and $m_2 = dm'_2$. Then the equation reads as $s_1 dm'_1 + x = s_2 dm'_2 + y$. Assume the equation has a solution n_0 for $x \neq kd + y$, where k is a fixed constant. Then we derive $n_0 \equiv x \pmod{d}$ and at the same time $n_0 \equiv y \pmod{d}$, a contradiction. So, the equation has only solutions if $x = kd + y$ (or vice versa). In this case, to find larger solutions one has to add numbers that are multiples of dm'_1 as well as of dm'_2 . On the other hand, adding $dm'_1 m'_2$ to a solution gives the next one. Since $(0, 0) \in \Delta_1 \times \Delta_2$ there are solutions at all and the intersection is infinite. The remaining argumentation that the intersection of L_1 and L_2 is of form $\langle dm'_1 m'_2, \Delta_3 \rangle$ with finite error set and $\Delta_3 = \{z_{x,y} \mid (x, y) \in \Delta_1 \times \Delta_2\}$ is along the lines of the first case.

Finally, since the empty word belongs to L_1 and L_2 and $0 \in \Delta_1 \cap \Delta_2$, we have $z_{0,0} = 0$ and, thus, the empty word belongs to the intersection and $0 \in \Delta_3$. If $a \in L_1 \cap L_2$ we have $1 \in \Delta_1 \cap \Delta_2$ by assumption. Moreover, in this case, the smallest solution for $x = y = 1$ is obtained for $s_1 = s_2 = 0$. That is, $z_{1,1} = 1$ and $1 \in \Delta_3$. \square

Now we are prepared to prove that certain unary languages are not intersection union free.

Lemma 3.4. *Let $x \geq 2$ be an integer. Then the language*

$$L = \{a\} \cup \{a^n \mid n \equiv 0 \pmod{x}\}$$

is not intersection–union free.

Proof. In contrast to the assertion assume that language L is described by some intersection–union-free expression r . We consider the expression tree of r whose nodes are attached with the languages described by the subtree rooted in the node. First, the expression tree is pruned as follows. Starting at the root, we travel along each branch until a node is labeled by a star or has a finite language attached. The rest of the branch is pruned. That is, we obtain a tree whose leaves are attached with a finite language or the star of some languages. Moreover, all inner nodes are attached with infinite languages and are labeled with either intersection or concatenation. The root is attached with L that includes the empty word. Since for intersection as well as for concatenation the empty word belongs to the resulting language if and only if it belongs to both operands, we know that all languages attached to the nodes include the empty word.

Now we turn to the language $\{a\}^*$. If a leaf is attached with $\{a\}^*$ we consider the predecessor node labeled either with intersection or concatenation. If it is concatenation, the resulting language is $\{a\}^*$ again, since the other operand includes the empty word. So, we replace the language attached to the concatenation by $\{a\}^*$ and prune the successors. If it is intersection with some language L' , the resulting language is L' . So, we replace the language attached to the intersection by L' and prune the successors. Since the root of the whole tree is not attached with $\{a\}^*$, after repeating this process as long as possible we obtain a tree where all nodes are not attached with $\{a\}^*$.

Considering the leaves labeled with a star we know that a does not belong to the language L' attached (otherwise it would be $\{a\}^*$), but L' includes the empty word and at least one non-empty word (otherwise it would be finite). By Lemma 2.3 there exists an $m \geq 1$ such that L' is obtained from a cofinite language stretched by m . Therefore, it is of form $\langle m, \Delta \rangle \cup \{\lambda\}$ with $\Delta = \{0\}$ and empty error set. Since $a \notin L'$, it satisfies the property that $a \in L'$ implies $1 \in \Delta$.

Continuing the evaluation from bottom to top, the next node is considered. It cannot be the intersection with a finite language, since all inner nodes are attached with infinite languages. Assume it is the concatenation with a finite language that necessarily includes the empty word. Then, by Lemma 3.1, the resulting language is again of form $\langle m', \Delta' \rangle \cup \{\lambda\}$ with $0 \in \Delta'$ and finite error set that satisfies the condition on a . Arguing inductively, Lemmas 3.2 and 3.3 show that finally the root of the tree is attached with a language of the form $\langle m'', \Delta'' \rangle \cup \{\lambda\}$ with $0 \in \Delta''$ and finite error set E'' that satisfies the condition on a . Since a belongs to L , it follows that $1 \in \Delta''$. So, Δ'' includes 0 and 1. Moreover, since apart from a language L includes only words whose lengths are congruent 0 modulo x , it follows that m'' has to be x . However, for each of these infinitely many words also the words extended by 1 belong to the language attached to the root. Since the error set is finite, it follows that L is not attached to the root, a contradiction. \square

Lemma 3.4 showed that there are unary regular languages that are not intersection–union free. In particular, the language $\{a\} \cup \{a^n \mid n \equiv 0 \pmod{x}\}$ is not intersection–union free, for any $x \geq 2$. On the other hand, if we trade the intersection for the complementation, Example 2.1 reveals that these languages are union free. In fact, in [20] it is shown that *all* unary regular languages are union free. So, the next corollary follows.

Corollary 3.5. *The unary intersection–union-free languages are strictly included in the unary union-free languages.*

Whether or not this inclusion is also for non-unary languages is currently an open problem. In general, both language families might be incomparable. However, the next result shows that the union-free languages do not capture all regular languages.

Theorem 3.6. *The regular language*

$$L = \{\lambda\} \cup \{awa \mid w \in \{a, b\}^*\} \cup \{bwb \mid w \in \{a, b\}^*\}$$

is not union free. Therefore, the union-free languages are strictly included in the regular languages.

Proof. Assume that L is described by some union-free expression and consider its expression tree. First we note that the root of the tree cannot be labeled with the star. In that case, for example, also the word $aabb$ would be described.

Second, let the root be labeled with concatenation and its successors be attached with languages L_1 and L_2 . Then the empty word belongs to L_1 as well as to L_2 . Therefore, in none of both languages there is a word beginning with an a and ending with a b , or vice versa. Moreover, if at least one word in L_1 begins with a , all non-empty words in L_2 end with a . This in turn implies that no word in L_1 begins with b and, thus, all words from L ending with b are in L_2 . This is a contradiction unless one of the languages coincides with $\{\lambda\}$. We conclude that the concatenation is useless and can be omitted since one of its operands equals its result.

Third, let the root be labeled with complementation. Then we consider the node following the complementation. If it is labeled with star a contradiction follows since the words a and b both do belong to the complement of L . Applying the star would describe also the words aa and bb that do belong to L but not to its complement. It remains to be shown that the concatenation of two languages L_1 and L_2 cannot result in the complement of L . Since λ does not belong to the complement but a and b do, exactly one of the languages includes the empty word, say L_1 . Then a and b are included in L_2 . This implies that none of the words in L_1 begins with a or b . Otherwise some word beginning and ending with a or b would be described that belongs to L but not to its complement. As before we conclude that L_1 coincides with $\{\lambda\}$ and that the concatenation is useless and can be omitted since one of its operands equals its result. So, the contradiction follows since the root of an expression tree that does not contain useless operations cannot be labeled with all possible operations concatenation, star, and complementation. \square

4. INTERSECTION–CONCATENATION-FREE EXPRESSIONS

This section is devoted to explore and relate the expressive capacities of intersection–concatenation-free expressions. The concatenation-free expressions have already been studied in [7], where it turned out that they do not capture all unary regular languages. We turn towards a characterization of unary languages described by intersection–concatenation-free expressions.

Lemma 4.1. *Let $m \geq 1$ be an integer. Every unary cofinite language stretched by m that includes the empty word is described by an intersection–concatenation-free expression.*

Proof. Let $L \subseteq \{a\}^*$ with $\lambda \in L$ be a language obtained from a cofinite language L_{cfin} that is stretched by m .

If $L_{\text{cfin}} = \{a\}^*$, then L is described by the intersection–concatenation-free expression $\{a^m\}^*$.

If $L_{\text{cfn}} \neq \{a\}^*$, we denote the length of the longest word not belonging to L_{cfn} by c . Now $p, q \geq 1$ are chosen relatively prime such that $p, q > c$. By Lemma 2.2, the language described by expression $(\{a^p\} \cup \{a^q\})^*$ includes all words whose lengths are greater than $pq - p - q$. On the other hand, since $p, q > c$, apart from λ it includes only words whose lengths are greater than c . Therefore, L_{cfn} is described by the expression $(\{a^p\} \cup \{a^q\})^* \cup L_{\text{fin},1} \cup L_{\text{fin},2}$, where $L_{\text{fin},1}$ is the finite language $\{a^n \mid c < n \leq pq - p - q\}$ and $L_{\text{fin},2}$ is the finite language including all words of L_{cfn} whose lengths are at most c . Finally, L is described by the intersection–concatenation-free expression $(\{a^{mp}\} \cup \{a^{mq}\})^* \cup L'_{\text{fin},1} \cup L'_{\text{fin},2}$, where $L'_{\text{fin},1}$ is the finite language $L_{\text{fin},1}$ stretched by m , and $L'_{\text{fin},2}$ is the finite language $L_{\text{fin},2}$ stretched by m . \square

In the sequel the family of all unary languages that are either finite or obtained from a cofinite language stretched by some $m \geq 1$ and including the empty word is denoted by \mathcal{U} . We are particularly interested in the union closure $\Gamma_{\cup}(\mathcal{U})$ of \mathcal{U} . Each language $L \in \Gamma_{\cup}(\mathcal{U})$ has a representation

$$\bigcup_{1 \leq i \leq k} L_i, \text{ where } k \geq 0 \text{ and } L_i \in \mathcal{U}.$$

Since any finite language is intersection–concatenation free, Lemma 4.1 and the trivial closure of the intersection–concatenation-free languages under union imply the next result.

Corollary 4.2. *Any language from $\Gamma_{\cup}(\mathcal{U})$ is intersection–concatenation free.* \square

Lemma 4.3. *The family of languages $\Gamma_{\cup}(\mathcal{U})$ is closed under intersection.*

Proof. Let $L_1 = L_{1,1} \cup L_{1,2} \cup \dots \cup L_{1,k}$ and $L_2 = L_{2,1} \cup L_{2,2} \cup \dots \cup L_{2,l}$ be two languages from the family $\Gamma_{\cup}(\mathcal{U})$. Recall, that all of the infinite sublanguages include the empty word, since they belong to \mathcal{U} . The intersection can be written as union of intersections, that is,

$$L_1 \cap L_2 = \bigcup_{\substack{1 \leq i \leq k \\ 1 \leq j \leq l}} L_{1,i} \cap L_{2,j}.$$

Now we consider the intersections $L_{1,i} \cap L_{2,j}$ separately. If at least one of both languages is finite the result is finite as well and belongs to \mathcal{U} . Next, we turn to the case where both languages to be intersected are infinite. That is $L_{1,i}$ is obtained from a cofinite language that is stretched by m_1 and $L_{2,j}$ is obtained from a cofinite language that is stretched by m_2 . Since both include the empty word, the empty word belongs to their intersection as well. Moreover, let ℓ_1 be the smallest integer such that all words of lengths $\{(\ell_1 + x)m_1 \mid x \geq 0\}$ belong to $L_{1,i}$, and similarly ℓ_2 for $L_{2,j}$. Then $L_{1,i}$ has a representation as $L_{\text{cfn},1} \cup L_{\text{fin},1}$, where $L_{\text{cfn},1} = \{\lambda\} \cup \{a^{(\ell_1+x)m_1} \mid x \geq 0\}$ and $L_{\text{fin},1}$ is a finite language. Similarly, $L_{2,j}$ has a representation as $L_{\text{cfn},2} \cup L_{\text{fin},2}$ with $L_{\text{cfn},2} = \{\lambda\} \cup \{a^{(\ell_2+x)m_2} \mid x \geq 0\}$ and $L_{\text{fin},2}$ finite. So, the intersection $L_{1,i} \cap L_{2,j}$ equals

$$(L_{\text{cfn},1} \cap L_{\text{cfn},2}) \cup (L_{\text{cfn},1} \cap L_{\text{fin},2}) \cup (L_{\text{fin},1} \cap L_{\text{cfn},2}) \cup (L_{\text{fin},1} \cap L_{\text{fin},2}),$$

where the last three of the joint languages are finite and, thus, belong to \mathcal{U} . Moreover, there exists a positive integer ℓ_3 such that the language $L_{\text{cfn},1} \cap L_{\text{cfn},2}$ has a representation as $\{\lambda\} \cup \{a^{(\ell_3+x)\text{lcm}(m_1,m_2)} \mid x \geq 0\}$. We conclude that $L_{\text{cfn},1} \cap L_{\text{cfn},2}$ is obtained from a cofinite language stretched by $\text{lcm}(m_1, m_2)$ and including the empty word and, thus, belongs to \mathcal{U} . So, the intersection $L_{1,i} \cap L_{2,j}$ belongs to $\Gamma_{\cup}(\mathcal{U})$ and, hence, $L_1 \cap L_2 \in \Gamma_{\cup}(\mathcal{U})$. \square

By definition the family of languages $\Gamma_{\cup}(\mathcal{U})$ is closed under union. We have already shown that it is closed under intersection. The next operation allowed in intersection–concatenation-free expressions is the star.

Lemma 4.4. *The family of languages $\Gamma_{\cup}(\mathcal{U})$ is closed under star.*

Proof. Let L be a language from the family $\Gamma_{\cup}(\mathcal{U})$. If it does not include a non-empty word, L^* is finite and, thus, belongs to \mathcal{U} . If L includes at least one non-empty word, then Lemma 2.3 says that there exists $m \geq 1$ such that L^* is obtained from a cofinite language that is stretched by m . Since $\lambda \in L^*$ we have $L^* \in \Gamma_{\cup}(\mathcal{U})$ in this case as well. \square

Now we are prepared to derive the characterization of the unary intersection–concatenation-free languages.

Theorem 4.5. *A unary language is intersection–concatenation free if and only if it belongs to the family $\Gamma_{\cup}(\mathcal{U})$.*

Proof. By Corollary 4.2, any language from $\Gamma_{\cup}(\mathcal{U})$ is intersection–concatenation free. In order to show the converse, let L be a language described by some intersection–concatenation-free expression r . The literals of r are finite subsets of words. They belong to \mathcal{U} and, thus, to $\Gamma_{\cup}(\mathcal{U})$. Then L is derived from the literals by finitely many applications of the operations union, star, and intersection. Since $\Gamma_{\cup}(\mathcal{U})$ is closed under union by definition and is closed under star and intersection by Lemmas 4.3 and 4.4, language L belongs to $\Gamma_{\cup}(\mathcal{U})$. \square

With the help of the characterization, now it can be shown that there are concatenation-free languages which are not intersection–concatenation free. First, we derive a class of languages that are not intersection–concatenation free.

Lemma 4.6. *Let $1 \leq x < y$ be integers. Then the language*

$$L = \{a^n \mid n \equiv x \pmod{y}\} \cup \{\lambda\}$$

is not intersection–concatenation free.

Proof. Contrarily assume that L is intersection–concatenation free. Then it belongs to $\Gamma_{\cup}(\mathcal{U})$ and has a representation of the form $L = L_1 \cup L_2 \cup \dots \cup L_k$ with languages L_i are either finite or obtained from a cofinite language stretched by some $m \geq 1$. Since L is infinite, there is some $1 \leq i \leq k$ such that L_i consists of infinitely many words whose lengths are congruent x modulo y . So, the differences between each two word lengths in L_i is a multiple of y . This implies that L_i is stretched by a multiple of y . Therefore, all word lengths in L_i are congruent 0 modulo y , a contradiction. \square

Since concatenation-free expressions allow the operations complementation and union, they can simulate the intersection as well. So, in general, the intersection–concatenation-free languages are included in the concatenation-free languages. We next turn to derive that this inclusion is strict. Results in [7] show that, for integers $0 \leq x < y$, the language $\{a^n \mid n \equiv x \pmod{y}\} \cup \{\lambda\}$ is concatenation free if and only if $x = 0$ or $x = \frac{y}{2}$.

Theorem 4.7. *The (unary) intersection–concatenation-free languages are strictly included in the (unary) concatenation-free languages.*

Proof. As mentioned before, the inclusion follows since the complementation together with the union can simulate the intersection. As witnesses for the strictness of the inclusion let $1 \leq x < y$ be two integers with $x = \frac{y}{2}$. Then the concatenation-free expression $r = \overline{\{a^x\}^* \cup \{a^y\}^*} \cup \{\lambda\}$ represents the language $L(r) = \{a^n \mid n \equiv x \pmod{y}\} \cup \{\lambda\}$. By Lemma 4.6, language $L(r)$ is not intersection–concatenation free. \square

We continue with one and a half incomparability results relating (intersection-)concatenation-free and (intersection-)union-free expressions.

Theorem 4.8. *The (intersection-)concatenation-free languages are incomparable with the intersection–union-free languages.*

Proof. Lemma 3.4 shows that the language $L = \{a\} \cup \{a^n \mid n \equiv 0 \pmod{x}\}$ is not intersection–union free, for all $x \geq 2$. It is described by the (intersection-) concatenation-free expression $\{a\} \cup \{a^x\}^*$. Conversely, for

integers $1 \leq x < y$, we consider languages $\{a^n \mid n \equiv x \pmod{y}\}$ which are described by the intersection–union-free expression $\{a^x\} \cdot \{a^y\}^*$. Results in [7] show that it is concatenation free if and only if $x = 0$ or $x = \frac{y}{2}$. So, for example, choosing $x = 1$ and $y = 3$ gives an intersection–union-free language that is not concatenation free. Since any intersection–concatenation-free language is concatenation free, the language is not intersection–concatenation free. \square

Since the intersection–union-free expressions $\{a^x\} \cdot \{a^y\}^*$ of the proof of Theorem 4.8 are also union free, the next corollary follows.

Corollary 4.9. *There is a union-free language that is not (intersection) concatenation free.*

Currently it is open whether there exists a non-unary regular language that is not concatenation free. The language $\{a^n b \mid n \geq 0\}$ is a candidate for that. To conclude this section we turn to show that the language is at least not described by any intersection–concatenation-free expression.

Theorem 4.10. *The language $L = \{a^n b \mid n \geq 0\}$ cannot be described by any intersection–concatenation-free expression.*

Proof. Assume that L is described by some intersection–concatenation-free expression and consider its expression tree. Every leaf is attached with a finite language. The only possibility to obtain an infinite language out of a finite one is to apply the star or to build the union with an infinite language. For the latter another infinite language is necessary. Starting at each leaf we travel along the path towards the root until a node attached with an infinite language L' is reached. If the node is labeled with a star we are faced with the following situation. If L' includes a word w from L , then it includes all words w^i , for $i \geq 2$, as well. If the star is applied to a language having this property, also the resulting language has this property. If such a language is joint with a finite one, the property survives the operation. The same is true when such a language is joint with another infinite language having the property. Applying the intersection results either in a finite language or in a language having this property. The latter can be seen as follows. If some $w \in L$ belongs to both languages then also all words w^i , for $i \geq 2$.

So, whenever a node in the tree is attached with an infinite language, the language has the property. Since L is infinite, the root of the tree is attached with a language having the property. But since $w \in L$ implies $w^i \notin L$, for $i \geq 2$, we obtain a contradiction. \square

5. RELATIONS WITH (INTERSECTION-)STAR-FREE EXPRESSIONS

Two of the remaining classes of expressions in question are intersection–star-free expressions and star-free expressions. While the latter received a lot of interest and are a well-investigated class of languages, the former has a quite simple characterization.

Lemma 5.1. *A language is described by an intersection–star-free expression if and only if it is finite.*

Proof. The literals of any intersection–star-free expression are finite subsets of words. The allowed operations are union, intersection, and concatenation. Applying each of these operations to finite languages results in finite languages again. \square

Corollary 5.2. *The (unary) intersection–star-free languages are strictly included in the (unary) star-free, (unary) intersection–concatenation-free, and (unary) intersection–union-free languages.*

Theorem 5.3. *The unary star-free languages are strictly included in the unary concatenation-free and unary union-free languages.*

Proof. A unary language is star free if and only if it is either finite or cofinite (see, for example, [4]). Since all finite languages are concatenation free and union free, and in both types of expressions the complementation is allowed, all cofinite languages are concatenation free and union free as well.

On the other hand, the concatenation-free expression $\overline{\{a^2\}^* \cup \{a^4\}^* \cup \{\lambda\}}$ describes the language $\{a^n \mid n \equiv 2 \pmod{4}\} \cup \{\lambda\}$ that is neither finite nor cofinite.

Example 2.1 gives the unary union-free language $\{a\} \cup \{a^{5 \cdot n} \mid n \geq 0\}$ that is neither finite nor cofinite. \square

For unary languages strict inclusions have been obtained of the star-free languages in the concatenation-free languages, of the intersection–union-free languages in the union-free languages, and of the star-free languages in the union-free languages. For the first two cases the relations for non-unary languages are open problems. However, in the latter case the strict inclusion of unary languages turns to incomparability for arbitrary alphabets.

Theorem 5.4. *The star-free languages are incomparable with the union-free languages.*

Proof. By Theorem 5.3 it remains to be shown that there is a star-free language which is not union free. Theorem 3.6 provides the non-union-free language

$$\{\lambda\} \cup \{awa \mid w \in \{a, b\}^*\} \cup \{bwb \mid w \in \{a, b\}^*\}.$$

It is described by the star-free expression $(\{a\} \cdot \bar{\emptyset} \cdot \{a\}) \cup (\{b\} \cdot \bar{\emptyset} \cdot \{b\}) \cup \overline{\{a, b\} \cdot \bar{\emptyset}}$. \square

A further incomparability result holds even for unary languages.

Theorem 5.5. *The (unary) star-free languages are incomparable with the (unary) intersection–concatenation-free languages.*

Proof. By Theorem 4.5 a unary language is intersection–concatenation free if and only if it belongs to the family $\Gamma_{\cup}(\mathcal{U})$. However, \mathcal{U} only includes finite languages and infinite languages that contain the empty word. This implies that also $\Gamma_{\cup}(\mathcal{U})$ includes only infinite languages containing the empty word. So, the star-free unary language $\{a\} \cdot \bar{\emptyset}$ is not intersection–concatenation free.

On the other hand, the intersection–concatenation-free language $\{aa\}^*$ is neither finite nor cofinite and, thus, is not star free. \square

6. SIMPLE EXPRESSIONS AND THEIR CLOSURES UNDER THE OMITTED OPERATION

Here, first we turn to complement the picture of what is known about simple subregular expressions. Simple concatenation-free expressions have been studied before in [9]. The simple union-free languages are a strict superset of the language family described by expressions as investigated in [12], where an expression also may consist the operations concatenation and star only, but where the literals are only letters from the alphabet. As a consequence, all languages described by expressions as studied in [12] are either infinite or contain a single word only.

Second, we investigate the closures of language families described by simple expressions under the operation that is omitted. In this way, we study the impact of allowing the operation only at the outermost level of the expressions.

6.1. Relations with simple expressions

Since even the intersection–star-free expressions describe finite languages only, the characterization of simple star-free languages is quite immediate.

Corollary 6.1. *A language is described by a simple star-free expression if and only if it is finite.*

The characterization of unary intersection–concatenation-free languages by the closure $\Gamma_{\cup}(\mathcal{U})$ obtained in Section 4 revealed that, in fact, the intersection is redundant. Every unary intersection–concatenation-free

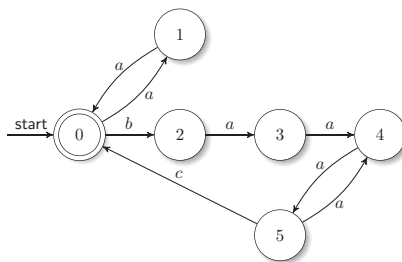


FIGURE 2. The minimal deterministic finite automaton accepting the language $L = \{baa, aa, ac\}^* \cap \{ba, aa, aac\}^*$.

language is described by an expression using the operations star and union only. However, it will turn out that this is no longer true for non-unary languages.

Each finite language is clearly simple concatenation free. It can be starred or not. In this way, expressions of the form $X_0 \cup X_1^* \cup \dots \cup X_n^*$ with all $X_i \subseteq \Sigma^*$ finite are obtained, which are simple concatenation free. The union of such expressions gives again an expression of this form. Moreover, the star of such an expression $(X_0 \cup X_1^* \cup \dots \cup X_n^*)^*$ is equal to $(\bigcup_{i=0}^n X_i)^*$. Since $\bigcup_{i=0}^n X_i$ is again a finite language, also the star of such an expression gives again an expression of this form. So, every simple concatenation-free language is described by some expression of the form above.

Lemma 6.2. *Let X be a finite set of words and u be the longest word in X . Then any word $w \in X^*$ with $|w| \geq 1$ can be factorized as $w = vv'$ with $v \in X^*$ and $1 \leq |v'| \leq |u|$.*

Proof. Since $w \in X^*$, it is the finite concatenation of words in X . That is, w can be written as $\lambda \cdot v_1 \cdot v_2 \cdots v_{k-1} \cdot v_k$, where $k \geq 1$ since $|w| \geq 1$, and $v_i \in X$ for $1 \leq i \leq k$. Therefore, $v = \lambda \cdot v_1 \cdot v_2 \cdots v_{k-1}$ belongs to X^* as well, and $|v_k| \leq |u|$ since u is the longest word in X . \square

Theorem 6.3. *The intersection-concatenation-free language*

$$L = \{baa, aa, ac\}^* \cap \{ba, aa, aac\}^*$$

cannot be described by any simple concatenation-free expression. Therefore, the simple concatenation-free languages are strictly included in the intersection-concatenation-free languages.

Proof. Assume that there is a simple concatenation-free expression describing L . Then L can be written as $X_0 \cup X_1^* \cup \dots \cup X_n^*$ with all $X_i \subseteq \{a, b, c\}^*$.

The minimal deterministic finite automaton depicted in Figure 2 accepts language L . The automaton accepts all words $baa(aa)^i ac$, for $i \geq 0$. So, there exists at least one $1 \leq i \leq n$ such that X_i^* includes infinitely many of these words. Let u be the longest word in X_i which must be non-empty. Then some word $baa(aa)^j ac$ with $j \geq |u|$ belongs to X_i^* . Now Lemma 6.2 says that $baa(aa)^j ac$ can be factorized into vv' with $v \in X_i^*$ and $1 \leq |v'| \leq |u|$. However, since $1 \leq |v'|$ we obtain for v a word of the form ba^k , for some $k \geq 1$. Since v does not belong to L it does not belong to X_i^* , a contradiction. \square

Now we turn to compare the remaining two types of simple subregular expressions.

As mentioned before, results in [7] show that the languages described by expressions $\{a^x\} \cdot \{a^y\}^*$ are concatenation free if and only if $x = 0$ or $x = \frac{y}{2}$. So, for example, choosing $x = 1$ and $y = 3$ gives a simple union-free language that is not even concatenation free. Similarly, Theorem 4.10 shows that language $\{a^n b \mid n \geq 0\}$ cannot be described by any intersection-concatenation-free expression. However, it is described by the simple union-free expression $\{a\}^* \cdot \{b\}$.

Corollary 6.4. *There are (unary) simple union-free languages that are not concatenation free.*

The converse is also true.

Lemma 6.5. *The simple concatenation-free languages $L_1 = \{a\}^* \cup \{b\}^*$ as well as $L_2 = \{a\} \cup \{aa\} \cup \{a^6\}^*$ are not simple union free.*

Proof. Assume that L_1 is described by some simple union-free expression and consider its expression tree. The words a and b belong to L_1 . If the root of the tree is labeled with a star, then a and b also belong to the language to which the star is applied. So, also the word ab not belonging to L_1 would be generated. We conclude that the root is not labeled with a star but with concatenation.

Now let L_1 be the concatenation of L' and L'' . One of the languages, say L' , has to include the word a . This implies that L'' does not include any word b^+ . So, all words b^+ belong to L' , which in turn implies that L'' does not include any word a^+ . Therefore, L'' must be the language $\{\lambda\}$. It follows that $L_1 = L'$ and the concatenation is useless. Since L_1 is infinite and, thus, no literal, we have a contradiction and L_1 is not simple union free.

Now assume that L_2 is described by some simple union-free expression and consider its expression tree. The word a belongs to L_2 . If the root of the tree is labeled with a star, then a also belongs to the language to which the star is applied. So, also the word aaa not belonging to L_2 would be generated. We conclude that the root is not labeled with a star but with concatenation.

Now let L_2 be the concatenation of L' and L'' . One of the languages, say L' , has to include the word a . Since the empty word belongs to L_2 it belongs to L' as well as to L'' . If L'' includes some word a^i where $i > 1$ is odd, then its concatenation with λ results in a word of odd length not belonging to L_2 . If, on the other hand, L'' includes some word a^i where $i > 1$ is even, then its concatenation with a results in a word of odd length not belonging to L_2 . We conclude that L'' consists solely of λ and possibly a . In order to generate the word $a^6 \in L_2$, either a^5 or a^6 must be included in L' . If $a \in L''$ then a^6 must not belong to L' . So, $a^5 \in L'$ which yields a contradiction since $\lambda \in L''$. Therefore, a does not belong to L'' and, thus, L'' must be the language $\{\lambda\}$. It follows that $L_2 = L'$ and the concatenation is useless. Since L_2 is infinite and, thus, no literal, we have a contradiction and L_2 is not simple union free. \square

Corollary 6.4 and Lemma 6.5 show the following incomparabilities.

Theorem 6.6. *The (unary) simple union-free languages are incomparable with the (unary) simple concatenation-free languages, with the (unary) intersection-concatenation-free languages, and with the (unary) concatenation-free languages.*

Though the unary simple union-free languages are included in the unary intersection-union-free languages which, in turn, are strictly included in the unary union-free languages, the edge that separates the unary language families is very small. In fact, the next theorem says that up to a finite set of words every unary regular language can be described by a simple union-free expression.

Theorem 6.7. *Let $L \subseteq \{a\}^*$ be a unary regular language. Then there is a simple union-free expression r such that $L = L(r) \cup L_{\text{fin}}$, where L_{fin} is a finite language.*

Proof. Let L be given by a complete deterministic finite automaton M . Since L is unary the state graph of M consists of an initial chain that ends in a cycle. Let k be the length of the cycle, F_1 be the set of accepting states on the chain, and F_2 be the set of accepting states on the cycle.

Now the simple union-free expression r is constructed as follows. Each word that is accepted by M with a state from F_1 is put into the finite language L_{fin} . For every state $s \in F_2$, the length c of the shortest path from the initial state to s is determined, and the word a^c is added to a finite set r_1 . So, all words that are accepted on the cycle are described by the simple union-free expression $r = r_1 \cdot \{a^k\}^*$, and $L = L(r) \cup L_{\text{fin}}$. \square

6.2. Closures of simple expressions

For all simple subregular expressions now we investigate the families of languages resulting from closing the family described under the omitted operation. From a structural perspective this means that the omitted operation is now allowed but at the outer level of the expressions only. Again, the question arises whether in this way all regular languages can be obtained or, if not, how the language families are related. For readability we use $\Gamma_{\cup}(\cdot, *)$ to denote the union closure of the simple union-free languages, and similarly for the remaining types of simple expressions.

First we examine the simple star-free languages.

Theorem 6.8. *Each language in $\Gamma_*(\cup, \cdot)$ is either finite or the star of a finite language.*

Proof. Since expressions of the form r^{**} can be simplified to r^* , and multiple applications of the operations \cup and \cdot to finite literals yield a finite language, each language in $\Gamma_*(\cup, \cdot)$ is either finite or the star of a finite language. \square

From the simplicity of the characterization of the last theorem it can be derived that $\Gamma_*(\cup, \cdot)$ is strictly included in the simple concatenation-free and simple union-free languages.

Theorem 6.9. *The family $\Gamma_*(\cup, \cdot)$ is strictly included in the simple concatenation-free and simple union-free languages.*

Proof. Since in simple concatenation-free as well as in simple union-free expressions the star may be applied to the finite literals, the inclusion follows immediately.

The infinite simple concatenation-free language $\{a\}^* \cup \{b\}^*$ as well as the infinite simple union-free language $\{a\} \cdot \{aa\}^*$ has no representation as star of finite languages. In the former case, every such finite language must include the words a and b and, thus, its star would include $\{a, b\}^*$. In the latter case, every such finite language must include the word a and, thus, its star would include $\{a\}^*$. \square

Though the family $\Gamma_*(\cup, \cdot)$ is strictly included in both non-trivial simple subregular language families, it is nevertheless incomparable with the star-free languages.

Theorem 6.10. *The family $\Gamma_*(\cup, \cdot)$ is incomparable with star-free languages.*

Proof. The unary language $\{aa\}^* \in \Gamma_*(\cup, \cdot)$ is neither finite nor cofinite and, thus, it is not star free. On the other hand, by the proof of Theorem 6.9 the language $\{a\}^* \cup \{b\}^*$ does not belong to $\Gamma_*(\cup, \cdot)$. However, it is described by the star-free expression $\overline{\emptyset} \cdot \{a\} \cdot \overline{\emptyset} \cup \overline{\emptyset} \cdot \{b\} \cdot \overline{\emptyset}$. \square

Let us turn to the family $\Gamma(\cup, *)$. In order to derive its relation with other families in question, we first provide witness languages not belonging to $\Gamma(\cup, *)$.

Lemma 6.11. *The languages*

$$L_1 = \{a\} \cdot \{b\}^* \cup \{b\} \cdot \{a\}^* \text{ and } L_2 = \{a, b\}^* \setminus (\{a\} \cdot \{a\}^*)$$

*do not belong to $\Gamma(\cup, *)$.*

Proof. Assume that L_1 belongs to the family $\Gamma(\cup, *)$ and consider its expression tree whose root is labeled with concatenation. So, L_1 is represented by the concatenation of some languages L' and L'' . If one of the languages L' or L'' is equal to $\{\lambda\}$, the concatenation is useless and can be omitted. Otherwise, there is at least one non-empty word in L' , say its first symbol is a . Then all non-empty words in L'' are of the form $\{b\}^+$. This, in turn, implies that none of the words in L' may have b as first letter. Therefore, all words from L_1 beginning with b belong to L'' . This implies that there is a word whose first symbol is a and who has a suffix of the form ba^+ belonging to the concatenation, a contradiction.

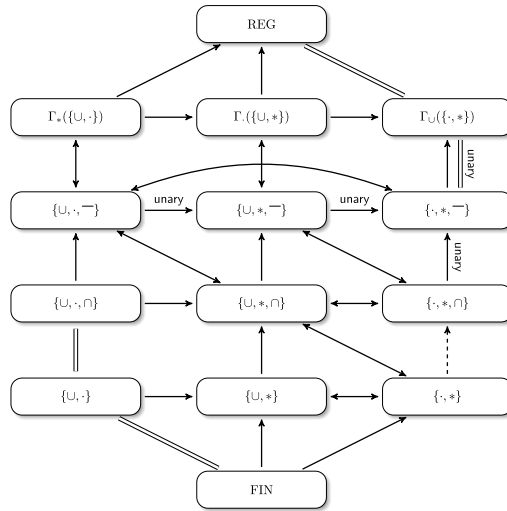


FIGURE 3. Main inclusion structure of language families described by different types of sub-regular expressions. For readability the vertices are labeled with the short form of the types or regular expressions. A double line indicates equality. Arrows with two heads indicate incomparabilities, the arrows with one head indicate strict inclusions. An arrow labeled unary indicates that the inclusion is for unary languages only. The dashed arrow indicates an inclusion that is not known to be strict. Some minor relations are not depicted, for example, the strict inclusion of the family $\Gamma_*(U, \cdot)$ in the families described by $\text{RE}(\Sigma, \Lambda, \{U, *\})$ and $\text{RE}(\Sigma, \Lambda, \{\cdot, *\})$.

Next, assume that the root of the expression tree is labeled by a star. Since a belongs to L_1 it belongs to the language to which the star is applied. Therefore, all words of the form a^* are generated as well, which is a contradiction.

So, we assume that the root is labeled by union. Then one of the joint languages includes infinitely many words of the form ba^+ . If this language would be generated by a star, words beginning with b and containing more than one letter b are generated, a contradiction. So, L_1 does not belong to $\Gamma.(U, *)$.

Next, assume that L_2 belongs to the family $\Gamma.(U, *)$ and consider its expression tree whose root is labeled with concatenation. So, L_2 is represented by the concatenation of some languages L'_1, L'_2, \dots, L'_k , for $k \geq 2$.

Since the empty word belongs to L_2 it belongs to all languages L'_i . Further, at least one of the languages, say L'_j , necessarily includes for infinitely many $m \geq 1$ a word of the form $vba^m bv'$, where $v, v' \in \{a, b\}^*$. Language L'_j has to be simple concatenation free. The discussion preceding Lemma 6.2 showed that any simple concatenation-free language has a representation as union of languages that are either finite or the star of a finite language. Knowing this, we can conclude that L'_j includes a unary word $w \in \{a\}^+$. Otherwise the infinitely many infixes a^m cannot be generated by a star applied to a finite language. But since the empty word belongs to all languages L'_i , the word w belongs to L_2 , a contradiction. So, L_2 does not belong to $\Gamma.(U, *)$. \square

Since the family $\Gamma_*(U, \cdot)$ is strictly included in the simple concatenation-free languages (Thm. 6.9), it is strictly included in the family $\Gamma.(U, *)$ as well. On the other hand, the next result gives the incomparability with concatenation-free languages.

Theorem 6.12. *The family $\Gamma.(U, *)$ is incomparable with concatenation-free languages.*

Proof. As mentioned before, results in [7] show that the languages described by expressions $\{a^x\} \cdot \{a^y\}^*$ are concatenation free if and only if $x = 0$ or $x = \frac{y}{2}$. For $x = 1$ and $y = 3$ we obtain a language that is not concatenation free but belongs to $\Gamma.(U, *)$.

On the other hand, Lemma 6.11 provides the language $L_2 = \{a, b\}^* \setminus (\{a\} \cdot \{a\}^*)$ that does not belong to $\Gamma(\cup, *)$. Writing L_2 as $\overline{\{a\}^* \cup \{\lambda\}}$, where the complement is built with respect to the alphabet $\{a, b\}$, shows that it is concatenation free. \square

The situation for the remaining family $\Gamma_{\cup}(\cdot, *)$ is different. In [11] it has been shown that every regular language can be represented as the union of simple union-free languages, where the literals are only letters. The idea is to transform a given regular expression by successively applying the equalities $(r_1 \cup r_2)^* = (r_1^* \cdot r_2^*)^*$ and $(r_1 \cup r_2) \cdot (r_3 \cup r_4) = r_1 \cdot r_3 \cup r_1 \cdot r_4 \cup r_2 \cdot r_3 \cup r_2 \cdot r_4$ in order to omit unions or to move them towards the outer level of the expression. In the end, the regular expression is a union of simple union-free expressions.

Corollary 6.13. *A language is regular if and only if it belongs to $\Gamma_{\cup}(\cdot, *)$.*

Theorem 6.14. *The family $\Gamma(\cup, *)$ and the union-free languages are strictly included in the family $\Gamma_{\cup}(\cdot, *)$.*

Proof. Lemma 6.11 provides a regular language not belonging to $\Gamma(\cup, *)$, and Theorem 3.6 provides a regular but not union-free language. \square

The main inclusion structure of the families in question is depicted in Figure 3.

7. CONCLUSIONS

We have studied the expressive capacity of different types of subregular expressions, where each type is obtained by either omitting one of the regular operations or replacing it by complementation or intersection. The power and limitations as well as relations with each other yield the hierarchical structure presented in Figure 3.

It turned out that the operation complementation is generally stronger than intersection. Considering the combination of operations, union together with star is stronger than union together with concatenation, and concatenation together with star is not weaker than union together with star. Closing the language families described by regular expressions with omitted operation under that operation gives a characterization of regular languages in case of union, while for the cases of concatenation and star incomparability results are obtained with the corresponding language families where the operation is replaced by complementation.

The relations labeled unary in Figure 3 are for unary languages only. The relations for general languages are open in these cases. Another open problem concerns the relation between $\text{RE}(\Sigma, \Lambda, \{\cdot, *\})$ and $\text{RE}(\Sigma, \Lambda, \{\cdot, *, \cup\})$. The former is included in the latter but it is open whether the inclusion is strict.

REFERENCES

- [1] R.S. Cohen and J.A. Brzozowski, Dot-depth of star-free events. *J. Comput. Syst. Sci.* **5** (1971) 1–16.
- [2] M. Fürer, The complexity of the inequivalence problem for regular expressions with intersection, in International Colloquium on Automata, Languages and Programming (ICALP 1980). Vol. 85 of *Lect. Notes Comput. Sci.* Springer, Berlin, Heidelberg (1980) 234–245.
- [3] M. Holzer and M. Kutrib, The complexity of regular(-like) expressions. *Int. J. Found. Comput. Sci.* **22** (2011) 1533–1548.
- [4] M. Holzer, M. Kutrib and K. Meckel, Nondeterministic state complexity of star-free languages. *Theor. Comput. Sci.* **450** (2012) 68–80.
- [5] H.B. Hunt, III, The Equivalence Problem for Regular Expressions with Intersections is not Polynomial in Tape. Technical Report TR 73-161, Department of Computer Science, Cornell University (1973).
- [6] S.C. Kleene, Representation of events in nerve nets and finite automata, in *Automata Studies*. Princeton University Press, NJ (1956) 3–42.
- [7] M. Kutrib and M. Wendlandt, Expressive capacity of concatenation freeness, in Implementation and Application of Automata (CIAA 2015). Vol. 9223 of *Lect. Notes Comput. Sci.* Springer, Cham (2015) 199–210.
- [8] M. Kutrib and M. Wendlandt, Expressive capacity of subregular expressions, in Non-Classical Models of Automata and Applications (NCMA 2016). Vol. 321 of books@ocg.at. Austrian Computer Society, Vienna (2016) 227–242.
- [9] M. Kutrib and M. Wendlandt, Concatenation-free languages. *Theor. Comput. Sci.* **679** (2017) 83–94.
- [10] R. McNaughton and S. Papert, Counter-Free Automata. *Research Monographs no. 65*. MIT Press, MA (1971).
- [11] B. Nagy, A normal form for regular expressions, in Supplemental Papers for DLT 2004. In Vol. 252 of CDMTCS. University of Auckland, Centre for Discrete Mathematics and Theoretical Computer Science (2004) 1–10.
- [12] B. Nagy, Union-free regular languages and 1-cycle-free-path automata. *Publ. Math. Debrecen* **68** (2006) 183–197.

- [13] H. Petersen, Decision problems for generalized regular expressions, in *Descriptional Complexity of Automata, Grammars and Related Structures (DCAGRS 2000)*. London, Ontario (2000) 22–29.
- [14] H. Petersen, The membership problem for regular expressions with intersection is complete in LOGCFL, in *Theoretical Aspects of Computer Science (STACS 2002)*. Vol. 2285 of *Lect. Notes Comput. Sci.* Springer, Berlin, Heidelberg (2002) 513–522.
- [15] K. Salomaa and S. Yu, Alternating finite automata and star-free languages. *Theor. Comput. Sci.* **234** (2000) 167–176.
- [16] M.-P. Schützenberger, On finite monoids having only trivial subgroups. *Inform. Control* **8** (1965) 190–194.
- [17] J. Shallit, The frobenius problem and its generalizations, in *Developments in Language Theory (DLT 2008)*. Vol. 5257 of *Lect. Notes Comput. Sci.* Springer, (2008) 72–83.
- [18] L.J. Stockmeyer, *The Complexity of Decision Problems in Automata Theory and Logic*. Ph.D. thesis, Massachusetts Institute of Technology, MA (1974).
- [19] L.J. Stockmeyer and A.R. Meyer, Word problems requiring exponential time, in *Symposium on Theory of Computing (STOC 1973)*. ACM Press, NY (1973) 1–9.
- [20] D. Werner, *Erweiterte union-free Sprachen über unärem Alphabet*. Bachelor’s thesis, Universität Giessen, Institut für Informatik, Germany (2013) (in German).