Descriptional Complexity and Operations – Two Non-classical Cases

Jürgen Dassow

HAL Id: hal-01657004
https://inria.hal.science/hal-01657004

Submitted on 6 Dec 2017

Descriptional Complexity and Operations – Two Non-Classical Cases

Jürgen Dassow

Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik,
PSF 4120, D-39016 Magdeburg, Germany
`dassow@iws.cs.uni-magdeburg.de`

Abstract. For a language family $\mathcal{L}$, a syntactic complexity measure K defined on languages of $\mathcal{L}$, a number $n \geq 1$, and an n-ary operation $\circ$ under which $\mathcal{L}$ is closed, we define $g_\circ^K(m_1, m_2, \ldots, m_n)$ as the set of all integers r such that there are n languages L_i, $1 \leq i \leq n$, with

$$K(L_i) = m_i \text{ for } 1 \leq i \leq n \text{ and } K(\circ(L_1, L_2, \ldots, L_n)) = r.$$

In this paper we study these sets for the operation union, catenation, star, complement, set-subtraction, and intersection and the measure number of accepting states (defined for regular languages) as well as for reversal, union, catenation, and star and the measures number of nonterminals, productions, and symbols (defined for context-free languages).
Moreover, we discuss the change of these sets if one restricts to finite languages, unary languages, and finite unary languages.

1 Introduction

The state complexity $\mathsf{sc}(L)$ of a regular language L is defined as the minimal number of states that are sufficient and necessary for a deterministic finite automaton to accept L. The study of the state complexity of regular languages is a central topic in theoretical computer science, but it has also large importance in applied fields. The first important results by Lupanov, Moore, Meyer, Fischer and others date back to the sixties and seventies, i.e., to the beginning of theoretical computer science.

In the last three decades the following problem was intensively investigated: Given a binary regularity preserving operation $\circ$ and two numbers m and n, determine the maximal number k (denoted by $f_\circ^{\mathsf{sc}}(m, n)$) such that there are languages L_m and L_n with $\mathsf{sc}(L_m) = m$, $\mathsf{sc}(L_n) = n$ and $\mathsf{sc}(L_m \circ L_n) = k$ (we have introduced the concept for binary operations, but the concept can be used for unary, ternary etc. operations as well). Summaries on the study of $f_\circ^{\mathsf{sc}}$ can be found in the papers [29] and [12].

As for other problems concerning the state complexity, one has noticed that the behaviour of the complexity under operations can considerably change if one restricts to finite or unary or finite unary languages. This can be seen from the following table.

Operation	regular	finite	reg. unary	fin. unary
Union	mn	$mn - (m+n)$	$\sim mn$	$\max\{m,n\}$
Intersection	mn	$mn - 3(m+n) + 12$	$\sim mn$	$\min\{m,n\}$
Complement	m	m	m	m
Kleene plus	$2^{m-1} + 2^{m-2} - 1$	m	$(m-1)^2$	m
Reversal	2^m	$O(2^{O(m)})$	m	m

We mention that there are also many papers where other subfamilies of the family of regular languages have been studied, e. g., star-free languages, union-free languages, languages closed under certain subword operations. Examples for such results can be found in [16] and [22].

There are also some results where instead of the maximal number $f_\circ^{\mathsf{sc}}$ the set $g_\circ^{\mathsf{sc}}$ of all numbers k such that there are languages L_m and L_n with $\mathsf{sc}(L_m) = m$, $\mathsf{sc}(L_n) = n$, and $\mathsf{sc}(L_m \circ L_n) = k$ is determined. We mention here three such results.

 – Complement: $g_C^{\mathsf{sc}}(m) = \{m\}$ for $m \geq 1$,
 – Union: ([20]) $g_\cup^{\mathsf{sc}}(m,n) = \{1, 2, \ldots, mn\}$ for $m \geq 2$ and $n \geq 2$
 – Kleene star: ([21]) $g_*^{\mathsf{sc}}(m) = \begin{cases} \{1,2\} & \text{for } m = 1 \\ \{1, 2, \ldots, 2^{m-1} + 2^{m-2}\} & \text{for } m \geq 2 \end{cases}$.

Obviously, the problem of the behaviour of syntactic measures of complexity under operations can be discussed in other cases, too, where one can change the complexity measure and/or the considered type of automaton. We mention here the following approaches.

The most natural extension of deterministic finite automata are nondeterministic finite automata. The operational behaviour of the (nondeterministic) state complexity is studied in [18] and summarized in [19].

The number of transitions is not of interest for complete deterministic finite automata, but the situation changes if one allows incomplete finite automata. In the papers [13] and [23], one can find results on the behaviour of the number of transitions under operations.

In order to cover XML structures one has to extend usual finite automata and comes to nested word automata or visibly pushdown automata or input-driven pushdown automata. The state complexity of these automata under operations is studied in the papers [1], [25], and [24].

Another natural extension of finite automata over strings is given by tree automata. There are also some results for their operational state complexity (see [26] and [27]).

In this paper we summarize results on the behaviour of some further non-classical measures of descriptional complexity under operations.

First we consider the number of accepting states instead of the number of (all) states. For a regular language L, the number $\mathsf{asc}(L)$ is defined as the minimal number of accepting states that are sufficient and necessary for a deterministic finite automaton to accept L. We mention two points of interest in this measure:

 – It was shown that, for two languages L_m and L_n with $\mathsf{sc}(L_m) = m$ and $\mathsf{sc}(L_n) = n$, the relation $\mathsf{sc}(L_m \cdot L_n) \leq m2^n - \mathsf{asc}(L_m) \cdot 2^{n-1}$ holds for

$m \geq 2$ and $n \geq 1$, and that the bound is optimal. For the Kleene-closure and the cut-operation, the complexity of the resulting language also depends on the number of accepting states (see [12] and [9]).

– The complexity of algorithms for the minimization of Büchi automata and for model checking based on Büchi automata depend on the number of accepting states of the Büchi automaton (see [5] and [10]).

We determine the sets $g_\circ^{\mathsf{asc}}$ for union, complement, set-subtraction, Kleene star, catenation, and intersection. Furthermore, we discuss variants of $g_\circ^{\mathsf{asc}}$ where we restrict the sets L_m and L_n to be finite sets or regular unary sets and or finite unary sets. The comparison shows that, for most operations, the situation is similar for arbitrary regular, finite sets, and regular unary sets, whereas finite unary sets show a completely different behaviour.

Some of these results were already published in [6].

Furthermore, we consider context-free languages and define their syntactic complexity as the minimal number of nonterminals or productions or symbols which is necessary to generate the language by context-free grammars. These measures were introduced and studied by Gruska in [14] and [15]. We summarize some results obtained in cooperation with Ralf Stiebe and Ronny Harbich (see [7], [8], and [17]) for the operational behaviour of these measures for arbitrary context-free languages under reversal, union, catenation, and star. Moreover, we add results for the case of finite, unary context-free, and finite unary languages. For the number of variables, there is a large difference which comes from the fact that the complexity for finite and unary context-free languages is bounded by one or two. For the number of productions, it seems that the difference between arbitrary and finite sets is essentially that we miss one or two "large" values.

2   Definitions and Notations

We assume that the reader is familiar with the basic notions of theory of automata and formal languages; for details we refer to [28]. Essentially, we give some notations and define the complexity measures of regular and context-free languages which are considered in this paper.

By $\mathrm{card}(M)$, we denote the cardinality of a set M. The empty word is denoted by λ. By $\mathbb{N}$, we denote the set of all positive integers. If L is a language, then we define the complement $C(L)$ of L as the set of all words $w \in V^*$ which are not contained in L, where V is the minimal set (with respect to inclusion) with $L \subseteq V^*$.

We specify a (deterministic) finite automaton as a tuple $\mathcal{A} = (Q, X, q_0, F, \delta)$ where Q and X are finite sets of states and inputs, respectively, $q_0 \in Q$ is a distinguished state (the initial state), F is a subset of Q (the set of accepting states), and δ is a function from $Q \times X$ into Q. The language accepted by $\mathcal{A}$ is denoted by $L(\mathcal{A})$.

For a finite automaton $\mathcal{A} = (Q, X, q_0, F, \delta)$ and a regular language L, we set

$$\texttt{asc}(\mathcal{A}) = \text{card}(F),$$
$$\texttt{asc}(L) = \min\{\texttt{asc}(\mathcal{A}) \mid L(\mathcal{A}) = L\}.$$

A context-free grammar is specified as a quadruple $G = (N, T, P, S)$ where N and T are two finite and disjoint sets, P is a finite subset of $N \times (v \cup N)^*$, and S is a distinguished element of N. The elements of N, T, and P are called nonterminals, terminals, and productions, respectively, and S is called the axiom. We write $A \to w$ instead of $(A, w) \in P$. By $L(G)$, we denote the language generated by G.

For a context-free grammar $G = (N, T, P, S)$, we set

$$\text{Var}(G) = \text{card}(N),$$
$$\text{Prod}(G) = \text{card}(P), and$$
$$\text{Symb}(G) = \sum_{A \to w \in P} (|w| + 2).$$

Let $K \in \{\text{Var}, \text{Prod}, \text{Symb}\}$. For a context-free language L, we set

$$K(L) = \min\{K(G) \mid L(G) = L\}.$$

For a language family $\mathcal{L}$, a syntactic complexity measure K defined on languages of $\mathcal{L}$, a number $n \geq 1$, and an n-ary operation $\circ$ under which $\mathcal{L}$ is closed, we define $g_\circ^K(m_1, m_2, \ldots, m_n)$ as the set of all integers r such that there are n languages L_i, $1 \leq i \leq n$, with

$$K(L_i) = m_i \text{ for } 1 \leq i \leq n \text{ and } K(\circ(L_1, L_2, \ldots, L_n)) = r.$$

If we additional require that the languages L_i, $1 \leq i \leq n$, are finite or unary or finite unary, we use the notations $g_\circ^{K,f}(m_1, m_2, \ldots, m_n)$, $g_\circ^{K,u}(m_1, m_2, \ldots, m_n)$, and $g_\circ^{K,f,u}(m_1, m_2, \ldots, m_n)$.

3 Number of Accepting States

In this section we only consider regular languages; therefore we omit the adjective "regular".

With respect to the measure number of accepting states, we consider the operations complement, union, set-substraction, catenation, star, and intersection. We start with complement.

Theorem 1. *The following relations hold for the operation complement:*

$$g_C^{\texttt{asc}}(m) = g_C^{\texttt{asc},u} \begin{cases} \{1\} & \text{for } m = 0 \\ \{0\} \cup \mathbb{N} & \text{for } m = 1 \\ \mathbb{N} & \text{for } m \geq 2 \end{cases}$$
$$g_C^{\texttt{asc},f}(m) = g_C^{\texttt{asc},f,u}(m) = \begin{cases} \{1\} & \text{for } m \in \{0\} \\ \mathbb{N} & \text{for } m \geq 2 \end{cases}.$$

We see that the only difference is that 0 is not in $g_C^{\text{asc},f}(1)$ and not in $g_C^{\text{asc},f,u}(1)$. This difference comes from the following observations:

- $\text{asc}(\emptyset) = 0$ holds, and $\text{asc}(L) \geq 1$ iff L is not empty.
- the complement of the empty set is the non-finite set V^* of all words (and $\text{asc}(V^*) = 1$).

We now give the results for union, set-subtraction, and star and compare them afterwards.

Theorem 2. *The following relations hold for the operation union:*

$$g_\cup^{\text{asc}}(m,n) = g_\cup^{\text{asc},u}(m,n) = \begin{cases} \{m\} & \text{for } n = 0, m \geq 0 \\ \{n\} & \text{for } m = 0, n \geq 0 \\ \mathbb{N} & \text{for } m \geq 1, \ n \geq 1 \end{cases},$$

$$g_\cup^{\text{asc},f}(m,n) = \begin{cases} \{m\} & \text{for } n = 0, m \geq 0 \\ \{n\} & \text{for } m = 0, n \geq 0 \\ \mathbb{N} & \text{for } m = n = 1 \\ \mathbb{N} \setminus \{1\} & \text{for } m \geq 1, \ n \geq 1, \ m+n \geq 3 \end{cases},$$

$$g_\cup^{\text{asc},f,u}(m,n) = \{\max\{m,n\}, \max\{m,n\}+1, \ldots, m+n\}.$$

Theorem 3. *The following relations hold for the operation set-subtraction:*

$$g_\setminus^{\text{asc}}(m,n) = g_\setminus^{\text{asc},u} = \begin{cases} \{0\} & \text{for } m = 0, n \geq 0 \\ \{m\} & \text{for } n = 0, m \geq 0 \\ \mathbb{N} \cup \{0\} & \text{for } m \geq 1, \ n \geq 1 \end{cases},$$

$$g_\setminus^{\text{asc},f}(m,n) = \begin{cases} \{0\} & \text{for } m = 0, n \geq 0 \\ \{m\} & \text{for } n = 0, m \geq 0 \\ \{0,1\} & \text{for } m = 1, n \geq 1, \\ \mathbb{N} \cup \{0\} & \text{for } m \geq 2, \ n \geq 1 \end{cases}$$

$$g_\setminus^{\text{asc},f,u}(m,n) = \{m, m-1, \ldots, m-n\}.$$

Theorem 4. *The following relations hold for the operation star:*

$$g_*^{\text{asc}}(m) = g_*^{\text{asc},u}(m) = \begin{cases} \{1\} & \text{for } m = 0 \\ \mathbb{N} & \text{for } m \geq 1 \end{cases},$$

$$g_*^{\text{asc},f}(m) = g_*^{\text{asc},f,u}(m) = \begin{cases} \{1\} & \text{for } m \in \{0,1\} \\ \mathbb{N} & \text{for } m \geq 2 \end{cases}.$$

First we mention that, for all these three operations union, set-subtraction, and star, there is no difference between the situations for arbitrary sets and arbitrary unary sets.

Moreover, for all these three operations, a difference between allowing arbitrary sets and restricting to finite sets only occurs for $m = 1$. In all cases, essentially, it comes from the following lemma.

Lemma 1. *Let L be a finite language. Then $\text{asc}(L) = 1$ if and only if L is prefix-free (i.e., no prefix of $w \in L$, which is different from w, is in L).*

If we now assume that $\mathtt{asc}(L \cup L') = 1$, then $L_m \cup L_n$ is prefix-free, and consequently L_m and L_n are prefix-free, too, which gives $\mathtt{asc}(L) = \mathtt{asc}(L') = 1$.

Moreover, if $\mathtt{asc}(L) = 1$ and thus L is prefix-free, we get that $L \setminus L'$ is prefix-free for all languages L'. Therefore, we get

$$\mathtt{asc}(L \setminus L') = \begin{cases} 0 & \text{if } L \subseteq L' \\ 1 & \text{otherwise} \end{cases}.$$

However, we see that there are differences between the situations for finite sets and finite unary sets at one hand and arbitrary unary sets and finite unary sets at the other hand. These differences come from the fact that, for a unary finite set L with n elements, we have $\mathtt{asc}(L) = n$, from which the statements for finite unary sets follow.

We now turn to catenation where we have no results – except for some trivial cases – for unary sets.

Theorem 5. *The following relations hold for the operation catenation:*

$$g^{\mathtt{asc}}_{\cdot}(m, n) = g^{\mathtt{asc},u}_{\cdot}(m, n) = \begin{cases} \{0\} & \text{for } \min\{m, n\} = 0 \\ \mathbb{N} & \text{for } m \geq 1, \ n \geq 1 \end{cases},$$

$$g^{\mathtt{asc},f}_{\cdot}(m, n) = \{0\} \text{ for } \min\{m, n\} = 0$$
$$g^{\mathtt{asc},f}_{\cdot}(m, n) \supseteq \{n + k \mid k \geq 0\} \text{ for } m \geq 2, n \geq 1$$
$$g^{\mathtt{asc},f,u}_{\cdot}(m, n) = \begin{cases} \{0\} & \text{for } \min\{m, n\} = 0 \\ \{k \mid m + n - 1 \leq k \leq m \cdot n\} & \text{for } m \geq 1, \ n \geq 1 \end{cases}.$$

Note that we do not know whether there is a difference for arbitrary and finite sets, since $g^{\mathtt{asc},f}_{\cdot}$ is not completely determined.

For all the preceding operations $\circ$, $g^{\mathtt{asc}}_{\circ}(m, n)$ was almost the set of all positive integers. This changes completely, if we consider intersection. Let L and L' be two regular sets accepted by the finite automata $\mathcal{A}$ and $\mathcal{A}'$, respectively. Then the standard construction of an automaton $\mathcal{B}$ accepting $L \cap L'$ gives an upper bound $\mathtt{asc}(\mathcal{A}) \cdot \mathtt{asc}(\mathcal{A}')$ for $\mathtt{asc}(L \cap L')$. Hence $g^{\mathtt{asc}}_{\cap}(m, n)$ contains only numbers $\leq m \cdot n$.

Theorem 6. *The following relation holds for the operation intersection: For $m \geq 0$ and $n \geq 0$,*

$$g^{\mathtt{asc}}_{\cap}(m, n) \supseteq \{(m - k)(n - l) + s \mid 0 \leq k \leq m, 0 \leq l \leq n, 0 \leq s \leq \min\{k, l\}\}.$$

We mention some easy consequences:
- $g^{\mathtt{asc}}_{\cap}(0, n) = g^{\mathtt{asc}}_{\cap}(m, 0) = \{0\}$ for $m \geq 0$ and $n \geq 0$,
- $m \cdot n \in g^{\mathtt{asc}}_{\cap}(m, n)$ for $m \geq 0$ and $n \geq 0$,
- for $m \geq 0$, we have $g^{\mathtt{asc}}_{\cap}(m, m) \supseteq \{r \mid 0 \leq r \leq \frac{4m}{9}\}$, i.e., a large section of small numbers is in $g^{\mathtt{asc}}_{\cap}(m, m)$.

For the unary case and intersection, we have the following statements:

Theorem 7. *The following relation holds for the operation intersection: For* $m \geq 0$ *and* $n \geq 0$,

$$g_\cap^{\mathsf{asc},u}(m,m) \supseteq \{(m-k)(m-k) + s \mid 0 \leq k \leq m, 0 \leq s \leq k\} \qquad (1)$$

and

$$g_\cap^{\mathsf{asc},f,u}(m,n) = \{0, 1, \ldots, \min\{m,n\}\}.$$

By the proof of (1), one can give an extension for $m \neq n$ and numbers k and l with $0 \leq k \leq m$ and $0 \leq l \leq n$, but one has to use in the formulation $\min\{m,n\}$ and $\min\{k,l\}$ which makes the formulae a little bit hard to read.

We have no useful result for $g_\cap^{\mathsf{asc},f}(m,n)$.

We have seen above that there are differences if we considered in general case or the unary case. Thus it is a natural question if there are further differences if we restrict the size of the underlying alphabet, i.e., if we consider the binary, ternary etc. case. We mention that all results presented above for arbitrary regular languages and finite languages already hold for alphabets with at least two letters. Therefore it is not necessary to distinguish by the size of the alphabet.

4   Syntactic Complexity Measures for Context-Free Languages

In this section we only consider context-free languages; therefore we omit the adjective "context-free.

We start with the remark that, $\mathrm{Var}(L) \leq 1$ for any finite language L (the set consisting of the words $w_1, w_2, \ldots, w_n$ is generated by a grammar with the rules $S \to w_1, S \to w_2, \ldots S \to w_n$) and $\mathrm{Var}(L) \leq 2$ for any unary context-free language ($L = \{a^{n_1}, a^{n_2}, \ldots, a^{n_s}\} \cup a^p\{a^{m_1}, a^{m_2}, \ldots, a^{m_t}\}$ is generated by a grammar with the rules $S \to a^{n_1}, S \to a^{n_2}, \ldots, S \to a^{n_s}, S \to S', S' \to a^p S', S' \to a^{m_2} S' \to a^{m_1}, \ldots, S' \to a^{m_t}$). Thus, the sets $g_\circ^{\mathrm{Var},f}(m,n)$ are not defined if $m \geq 2$ or $n \geq 2$, and the sets $g_\circ^{\mathrm{Var},u}(m,n)$ are not defined if $m \geq 3$ or $n \geq 3$.

We now present the results for the reversal operation.

Theorem 8. *For* $K \in \{\mathrm{Var}, \mathrm{Prod}, \mathrm{Symb}\}$ *and all permissible* $m \in \mathbb{N} \cup \{0\}$,

$$g_R^K(m) = \{m\}, \ g_R^{K,f}(m) = \{m\}, \ g_R^{K,u}(m) = \{m\}, \ g_R^{K,f,u}(m) = \{m\}.$$

From commutativity of union and Theorem 8, it follows that, for the measures $K \in \{\mathrm{Var}, \mathrm{Prod}, \mathrm{Symb}\}$,

$$g_\cup^K(m,n) = g_\cup^K(n,m) \text{ and } g_\cdot^K(m,n) = g_\cdot^K(n,m)$$

and the corresponding relations also hold if we restrict to finite languages, unary languages, and finite unary languages. Therefore we can assume without loss of generality that $m \geq n$, if we discuss union or product.

We have the following results concerning operations and the number of variables.

arbitrary	$g_\cup^{\mathrm{Var}}(m,n) = \{1, 2, \ldots, m+n+1\}$ for $m \geq 1, n \geq 1$
finite	$g_\cup^{\mathrm{Var},f}(1,0) = g_\cup^{\mathrm{Var},f}(0,1) = g_\cup^{\mathrm{Var},f}(1,1) = \{1\}, \ \ g_\cup^{\mathrm{Var},f}(0,0) = \{0\}$
unary	$g_\cup^{\mathrm{Var},u}(m,n) = \{1, 2\}$ for $1 \leq m \leq 2, 1 \leq n \leq 2$ $g_\cup^{\mathrm{Var},u}(1,0) = g_\cup^{\mathrm{Var},u}(0,1) = \{1\}, g_\cup^{\mathrm{Var},u}(0,0) = \{0\}$
finite unary	$g_\cup^{\mathrm{Var},f,u}(1,0) = g_\cup^{\mathrm{Var},f,u}(0,1) = g_\cup^{\mathrm{Var},f,u}(1,1) = \{1\}, g_\cup^{\mathrm{Var},f,u}(0,0) = \{0\}$

Fig. 1. Behaviour of the number of variables under union

Theorem 9. *The behaviour of* Var *under union is shown in Figure 1.*

Theorem 10. *The behaviour of* Var *under catenation is shown in Figure 2.*

arbitrary	$g_{\cdot}^{\mathrm{Var}}(m,n) \supseteq \{1\} \cup \{\max\{m,n\}, \max\{m,n\}+1, \ldots, m+n+1\}$ for $m \geq 1, n \geq 1$
finite	$g_\cup^{\mathrm{Var},f}(1,1) = \{1\}, g_{\cdot}^{\mathrm{Var},f}(1,0) = g_{\cdot}^{\mathrm{Var},f}(0,1) = g_\cup^{\mathrm{Var},f}(0,0) = \{0\}$
unary	$g_{\cdot}^{\mathrm{Var},u}(m,n) = \{1, 2\}$ for $1 \leq m \leq 2, 1 \leq n \leq 2$ $g_{\cdot}^{\mathrm{Var},u}(1,0) = g_{\cdot}^{\mathrm{Var},u}(0,1) = g_{\cdot}^{\mathrm{Var},u}(0,0) = \{0\}$
finite unary	$g_\cup^{\mathrm{Var},f,u}(1,1) = \{1\}, g_{\cdot}^{\mathrm{Var},f,u}(1,0) = g_{\cdot}^{\mathrm{Var},f,u}(0,1) = g_\cup^{\mathrm{Var},f,u}(0,0) = \{0\}$

Fig. 2. Behaviour of the number of variables under catenation

Theorem 11. *The behaviour of* Var *under Kleene star is shown in Figure 3.*

We see that there is a large difference between arbitrary regular languages on the one side and finite or unary sets on the other hand, but the difference only originates from the restricted domain of $g_\circ^{\mathrm{Var},f}$ and $g_\circ^{\mathrm{Var},u}$. Between finite and finite unary sets, there is no difference.

We now consider the number of productions.

Theorem 12. *Let $m \geq n \geq 2$.*
  i) The number 1 and all numbers k with $k > m+n+2$ are not in $g_\cup^{\mathrm{Prod}}(m,n)$.
  ii) If $n \geq 7$, then $\{k \mid 6 \leq k \leq m+n+2\} \subseteq g_\cup^{\mathrm{Prod}}(m,n)$.
  iii) If $n \in \{5, 6\}$, then $\{k \mid 4 \leq k \leq m+n+2\} \subseteq g_\cup^{\mathrm{Prod}}(m,n)$.
  iv) If $n = 4$, then $\{2\} \cup \{k \mid 4 \leq k \leq m+n+2\} \subseteq g_\cup^{\mathrm{Prod}}(m,n)$.
  v) If $n \in \{2, 3\}$, then $\{k \mid 2 \leq k \leq m+n+2\} = g_\cup^{\mathrm{Prod}}(m,n)$.

arbitrary	$g_*^{\mathrm{Var}}(n) = \{1, 2, \ldots, n+1\}$ for $n \geq 1$
finite	$g_*^{\mathrm{Var},f}(1) = g_*^{\mathrm{Var},f}(0) = \{1\}$,
unary	$g_*^{\mathrm{Var},u}(1) = g_*^{\mathrm{Var},u}(2) = \{1\}$, $g_*^{\mathrm{Var},u}(0) = \{1\}$
finite unary	$g_*^{\mathrm{Var},f,u}(1) = g_*^{\mathrm{Var},f,u}(0) = \{1\}$,

Fig. 3. Behaviour of the number of variables under star

Theorem 13. *i) For all numbers m with $m \geq 2$, the number 1 and all numbers k with $k > m+2$ are not in $g_\cup^{\mathrm{Prod}}(m,1)$.*
ii) For $m \geq 2$, we have $\{k \mid n \leq k \leq m+2\} \subseteq g_\cup^{\mathrm{Prod}}(m,1)$.
For $m \geq 5$, the relation $n - 1 \in g_\cup^{\mathrm{Prod}}(m,1)$ holds.
For $n \geq 6$, we have $n - 2 \in g_\cup^{\mathrm{Prod}}(m,1)$.
Moreover, $g_\cup^{\mathrm{Prod}}(1,1) = \{1,2\}$ is valid.

We notice that the open problems concern only small numbers, i. e., it is open whether or not the following relations hold:
- $2, 3 \in g_\cup^{\mathrm{Prod}}(m,1)$ for $m \in \{4,5\}$ and $2, 3, \ldots, m-2 \in g_\cup^{\mathrm{Prod}}(m,1)$ for $m \geq 6$,
- $3 \in g_\cup^{\mathrm{Prod}}(m,4)$ for $m \geq 4$,
- $2, 3 \in g_\cup^{\mathrm{Prod}}(m,5)$ for $m \geq 5$,
- $2, 3, 4 \in g_\cup^{\mathrm{Prod}}(m,6)$ for $m \geq 6$,
- $2, 3, 4, 5 \in g_\cup^{\mathrm{Prod}}(m,n)$ for $m \geq n \geq 7$.

Theorem 14. *i) For all numbers $m \geq 1$ and $n \geq 1$, the number 1 and all numbers k with $k > m+n$ are not in $g_\cup^{\mathrm{Prod},f}(m,n)$.*
ii) If $m \geq n \geq 1$, then $\{m, m+1, \ldots, m+n\} \subseteq g_\cup^{\mathrm{Prod},f}(m,n)$.
iii) If $m \geq n \geq 3$, then $\{n, n+1, \ldots m+n\} \subseteq g_\cup^{\mathrm{Prod},f}(m,n)$.
iv) If $m \geq 6$ and $m \geq n \geq 4$, then $\{6, 7, \ldots, m+n\} \subseteq g_\cup^{\mathrm{Prod},f}(m,n)$.

Theorem 15. *i) For all numbers $m \geq 1$ and $n \geq 1$, the number 1 and all numbers k with $k > m+n+2$ are not in $g_\cup^{\mathrm{Prod},u}(m,n)$.*
ii) If $m \geq n \geq 1$, then $\{m, m+1, \ldots m+n\} \subseteq g_\cup^{\mathrm{Prod},u}(m,n)$.
iii) If $m \geq n \geq 3$, then $\{n, n+1, \ldots m+n\} \subseteq g_\cup^{\mathrm{Prod},u}(m,n)$.

The essential differences between the cases of arbitrary, finite, and unary sets are:
- If we restrict to finite sets, then $m+n+1$ and $m+n+2$ are not in $g_\cup^{Prod,f}(m,n)$ and not in $g_\cup^{Prod,f,u}(m,n)$.
- If we restrict to the unary case we only know that the numbers $k \geq \min\{m,n\}$ are in $g_\cup^{Prod,u}(m,n)$ and $g_\cup^{Prod,f,u}(m,n)$; for "small" numbers we miss constructions.

For the star operation, we have the following results.

Theorem 16. *The following relations hold:*
i) $g_*^{\mathrm{Prod}}(0) = g_*^{\mathrm{Prod},f}(0) = \{1\}$,
ii) $g_*^{\mathrm{Prod}}(1) = g_*^{\mathrm{Prod},f}(1) = \{1,2\}$,
iii) For $n \geq 2$, $g_*^{\mathrm{Prod}}(n) = \{2,3,\ldots,n{+}2\}$ *and* $g_*^{\mathrm{Prod},f}(n) = \{2,3,\ldots,n{+}1\}$.

We see that the only difference between arbitrary context-free and finite sets is that $n+2$ is not contained in $g_*^{\mathrm{Prod},f}(n)$.
We do not have non-trivial results for $g_*^{\mathrm{Prod},u}$ and $g_*^{\mathrm{Prod},f,u}$.

We now discuss the concatenation and restrict to the general case, because all our proofs require infinite languages and languages over an alphabet with at least two letters (i.e., we cannot present results on $g.^{\mathrm{Prod},f}$, $g.^{\mathrm{Prod},u}$, and $g.^{\mathrm{Prod},f,u}$).

Theorem 17. *i) For all numbers* $m \geq n \geq 1$, *the number 0 and all numbers* k *with* $k \geq m+n+1$ *are not in* $g.^{\mathrm{Prod}}(m,n)$. *Moreover, if* $m \geq 2$, *then* $m+2 \notin g.^{\mathrm{Prod}}(m,1)$ *and* $1 \notin g.^{\mathrm{Prod}}(m,n)$.
ii) We have $\{1\} = g.^{\mathrm{Prod}}(1,1)$.
iii) We have $\{2,3,4,5\} = g.^{\mathrm{Prod}}(2,2)$ *and* $\{m,m{+}1,m{+}2,m{+}3\} \subseteq g.^{\mathrm{Prod}}(m,2)$ *for* $m \geq 2$.
iv) We have $\{m+n-1,m+n,m+n+1\} \subseteq g.^{\mathrm{Prod}}(m,n)$ *for* $m,n \in \{3,4\}$.
v) We have $\{n+2,n+3,\ldots m+n+1\} \subseteq g.^{\mathrm{Prod}}(m,n)$ *for* $m \geq n \geq 5$.

With respect to the number of symbols the situation is not very clear for small numbers n and m. We only give the results for "large" numbers; for a proof and further facts we refer to [17].

Theorem 18. *i) For all numbers* $m \geq 23$ *and* $n \geq 23$, *we have*

$$\{k \mid k \geq m+n+7\} \cap g_\cup^{\mathrm{Symb}}(m,n) = \emptyset \ \ and$$
$$\{23,24,\ldots,n\} \cup \{n+3,n+4,\ldots,m+n-2\} \cup \{m+n+6\} \subseteq g_\cup^{\mathrm{Symb}}(m,n).$$

ii) For all numbers $m \geq 8$ *and* $n \geq 5$ *with* $m \geq n$ *we have*

$$\{k \mid k \geq m+n+5\} \cap g.^{\mathrm{Symb}}(m,n) = \emptyset \ \ and$$
$$\{n+4,n+5\ldots,m+n-2\} \cup \{m+n+4\} \subseteq g.^{\mathrm{Symb}}(m,n).$$

iii) For all numbers $m \geq 10$, *we have*

$$\{k \mid k \geq m+7\} \cap g_*^{\mathrm{Symb}}(m) = \emptyset \ \ and \ \{10,11,\ldots,m\} \cup \{m+6\} \subseteq g_*^{\mathrm{Symb}}(m).$$

5   Conclusion

We have presented a summary of result concerning the operational complexity of the number of accepting states for regular languages and of the number of nonterminals, productions, and symbols for context-free languages.

For the number of accepting states and the operations union, set-subtraction, complement, and star, the results are complete, and we see that there are –

essentially – no differences between arbitrary, finite, and unary sets. However, the finite unary sets behave completely differently. With respect to catenation, the situation can be the same, but, for a proof, a complete determination of $g_{\cdot}^{\mathtt{asc},f}$ is necessary (and missing at present). For the intersection, we have not enough information in order to make a statement on the comparison.

The situation is different for the syntactic measures of context-free languages. Concerning the number of nonterminals, we have a difference between arbitrary context-free languages and the restricted versions, but it comes from the very limited domain (of the variables m and n in the case of restrictions). If we restrict to "large" arguments (say $m \geq n \geq 50$, which can be justified by practical reasons), we have a good situation for the number of productions with respect to union and star since the difference between arbitrary context-free sets and finite sets is only in two or one "large" values. For the comparison of unary and finite unary languages, we need more information.

In order to get a more complete picture, it is necessary to determine the sets $g_{\cdot}^{\mathrm{Prod},f}$, $g_{\cdot}^{\mathrm{Prod},u}$, and all sets with a restriction and the measure Symb.

Finally, we mention that it remains open to determine the sets $g_{\circ}^{\mathtt{asc}}$ and their restricted versions for further operations as reversal (L^R) or squaring (L^2), quotients, etc. as well as the sets $g_{\circ}^{K}$ with $K \in \{\mathrm{Var}, \mathrm{Prod}, \mathrm{Symb}\}$ for operations as squaring, quotients, etc.

References

1. Alur, R., Madhusudan, P.: Visibly pushdown languages. In: Proc. ACM Symposium on Theory of Computing, pp. 202–211 (2004)
2. Brzozowski, J.: Quotient complexity of regular languages. Journal of Automata, Languages, and Combinatorics 15, 71–89 (2010)
3. Brzozowski, J., Jirásková, G., Li, B.: Quotient complexity of ideal languages. In: López-Ortiz (ed.) LATIN 2010, LNCS 6034, pp. 208–221 (2010)
4. Brzozowski, J., Jirásková, G., Zou, Ch.: Quotient complexity of closed languages. Theory of Computing Systems 54, 277–292 (2014)
5. Champarnaud, J.-M., Coulon, F.: Büchi automata reduction by means of left and right trace inclusion preorder. Manuscript, 2004
6. Dassow, J.: On the number of accepting states of finite automata. Journal of Automata, Languages, and Combinatorics 21, 55–67 (2016)
7. Dassow, J., Stiebe, R.: Nonterminal complexity of some operations on context-free languages. fundamenta Informaticae 83, 35-49 (2008)
8. Dassow, J., Harbich, R.: Descriptional complexity of union and star on context-free languages. Journal of Automata, Languages, and Combinatorics 17, 123–143 (2012)
9. Drewes, F., Holzer, M., Jakobi, S., van der Merwe, B.: Tight bounds for cut-operations on deterministic finite automata. In: Durand-Lose, J., Nagy, B. (eds.): Machines, Computations, Universality, LNCS 9288, Springer-Verlag, Heidelberg, pp. 45–60 (2015)
10. Edelkamp, St., Jabbar, Sh.: Large scale directed model checking LTL. In: Valmer, A. (ed.) Model Checking Software - SPIN, LNCS 3925, Springer-Verlag, Berlin, pp. 1–18 (2006)

11. Evangelista, S., Kristensen, L. M.: Hybrid on-the-fly LTL model checking with the sweep-line method. In: Haddad, S., Pomello, L. (eds.) Applications and Theory of Petri Nets, LNCS 7347, Springer-Verlag, Berlin, pp. 248–267 (2012)

12. Gao, Y., Moreira, N., Reis, R., Yu, Sh.: A review on state complexity of individual operations. Techn. report series DCC-2011-08, Version 1.1 September 2012, University of Porto, Faculty of Sciences, Department of Computer Science (2012)

13. Gao, Y., Salomaa, K., Yu, Sh.: Transition complexity of incomplete dfas. Fundamenta Informaticae 110, 143–158 (2011)

14. Gruska, J.: Some Classifications of Context-Free Languages. Information and Control 14, 152-û179 (1969)

15. Gruska, J.: On the Size of Context free Grammars. Kybernetika 8, 213-û218 (1972)

16. Han, Y.-S., Salomaa, K.: State complexity of basic operations on suffix-free regular languages. Theoretical Computer science 410, 2537–2548 (2009)

17. Harbich, R., Regel- und Symbolkomplexität kontextfreier Sprachen unter ausgewählten Operationen. Dissertation (2018)

18. Holzer, M., Kutrib, M.: State complexity of basic operations on nondeterministic automata. In: Champernaud, J.-M., Maurel, D. (eds.)Implementation and Application of Automata, LNCS 2608, pp. 148–157 (2003)

19. Holzer, M., Kutrib, M.: nondeterministic finite automata – recent results on the descriptional and computational complexity. Internat. Journal of Foundations of Computer Science 20, 563–580 (2009)

20. Hricko, M., Jirásková, G., Szabari, A.: Union and intersection of regular languages and descriptional complexity. In: Mereghetti, C., Palano, B., Pighizzini, G., Wotschke, D. (eds.) Proc. 7th Intern. Workshop of Descriptional Complexity of Formal Systems, University of Milano, pp. 170–181 (2005)

21. Jirásková, G.: On the state complexity of complements, stars, and reversals of regular languages. In: Ito, M., Toyama, M. (eds.) Developments in Language Theory, LNCS 5257, Springer-Verlag, Berlin, pp. 431–442 (2008)

22. Jirásková, G., Krausová, M.: Complexity in prefix-free regular languages. Electron. Proc. Theor. Comput. Sci. 31, 197–204 (2010)

23. Maia, E., Moreira, N., Reis, R.: Incomplete operational transition complexity of regular languages. Information and Computation 244, 1–22 (2015)

24. Okhotin, A., Salomaa, K.: State complexity of operations on input-driven pushdown automata. Journal of Computer and System Sciences 86, 207–228 (2017)

25. Piao, X., Salomaa, K.: Operational state complexity of nested word automata. Theoretical Computer Science 410, 3250-3260 (2009)

26. Piao, X., Salomaa, K.: State complexity of the concatenation of regular tree automata. Theoretical Computer science 429, 273–281 (2012)

27. Piao, X., Salomaa, K.: State complexity of projection and quotient of unranked trees. In: Kutrib, M., Moreira, N., Reis, R. (eds.) Descriptional Complexity of Formal Systems, LNCS 7386, pp. 280–293 (2012)

28. Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages. Vol. I –III, Springer-Verlag, Berlin (1997)

29. Yu, Sh.: State complexity of regular languages. Journal of Automata, Languages and Combinatorics 6 221–234 (2001)