Theoretical Computer Science 798 (2019) 109–125

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

On the cover complexity of finite languages [☆]

Stefan Hetzl [*], Simon Wolfsteiner

TU Wien, Institute of Discrete Mathematics and Geometry, Wiedner Hauptstraße 8–10, 1040 Vienna, Austria

ARTICLE INFO

Article history:
Received 3 December 2018
Received in revised form 20 March 2019
Accepted 5 April 2019
Available online 23 May 2019

Keywords:
Context-free grammar
Finite language
Descriptional complexity
Cover complexity
Language operation

ABSTRACT

We consider the notion of cover complexity of finite languages on three different levels of abstraction. For arbitrary cover complexity measures, we give a characterisation of the situations in which they collapse to a bounded complexity measure. Moreover, we show for a restricted class of context-free grammars that its grammatical cover complexity measure w.r.t. a finite language L is unbounded and that the cover complexity of L can be computed from the exact complexities of a finite number of covers $L' \supseteq L$. We also investigate upper and lower bounds on the grammatical cover complexity of the language operations intersection, union, and concatenation on finite languages for several different types of context-free grammars. One of the lower bound results is based on a new class of cover-incompressible languages.

1. Introduction

The grammatical complexity of a formal language in the classical sense is the complexity of a minimal grammar generating this language. Depending on the type of grammar and the notion of complexity, one obtains a variety of different grammatical complexity measures. The study of the grammatical complexity of context-free languages can be traced back to [1], where, among other things, it was shown that context-free definability with n nonterminals forms a strict hierarchy. This line of research has been continued in [2–5], where, among others, the number of productions of a grammar has been considered as complexity measure. In [6], a theory of the grammatical complexity of finite languages in terms of production complexity was initiated by giving a relative succinctness classification for various kinds of context-free grammars. Investigations along these lines have been continued in, e.g., [7–12].

We are interested in the cover complexity of a finite language L, i.e., the minimal number of productions of a grammar G such that $L(G)$ is finite and $L(G) \supseteq L$. Note that this condition is similar to (but different from) the one imposed on cover automata [13,14]: there, an automaton $\mathcal{A}$ is sought such that $L(\mathcal{A}) \supseteq L$, but in addition it is required that $L(\mathcal{A}) \setminus L$ consists only of words longer than any word in L. Our interest in this problem is primarily motivated by applications in proof theory. As shown in [15], there is an intimate relationship between a certain class of formal proofs (those with Π_1-cuts) in first-order predicate logic and a certain class of grammars (totally rigid acyclic tree grammars). In particular, the number of production rules in the grammar characterises the number of certain inference rules in the proof. This relationship has been exploited for a number of results in proof theory and automated deduction [16–19]. In particular, [20,21] shows a non-trivial lower bound on the complexity of cut-introduction. The interest in such a result is partially motivated by the experience that the length of proofs with cuts is notoriously difficult to control (for propositional logic this is considered

[☆] Supported by the Vienna Science Fund (WWTF) project VRG12-004 and the Austrian Science Fund (FWF) project P25160.

[*] Corresponding author.
E-mail addresses: stefan.hetzl@tuwien.ac.at (S. Hetzl), simon.wolfsteiner@tuwien.ac.at (S. Wolfsteiner).

https://doi.org/10.1016/j.tcs.2019.04.014
0304-3975/© 2019 Elsevier B.V. All rights reserved.

the central open problem in proof complexity [22]). The combinatorial centre of this result is the construction of a sequence of finite word languages which are incompressible in the sense of the cover formulation of grammatical complexity.

In [23], the grammatical cover complexity of finite languages has also been investigated as part of a relative succinctness classification. There, the authors considered several different complexity measures (among others, exact and cover complexity) for finite languages and related them according to four different relations. Furthermore, in [24], it was shown that the minimal cover problem for acyclic regular grammars with a fixed bound on the number of nonterminals is NP-complete. The minimal cover problem is defined as follows: given a finite language L and a non-negative integer k, is there an acyclic regular grammar G such that G has at most k productions and satisfies $L(G) \supseteq L$. The computational complexity of this problem for an arbitrary number of nonterminals is still open.

In this paper, we investigate the notion of cover complexity of finite languages on three different levels. First, in Section 3, we consider the cover complexity from an abstract point of view for arbitrary complexity measures and we characterise the situations in which it collapses to a bounded measure. Secondly, in Section 4, we consider the cover complexity of a finite language as the minimal number of productions a grammar needs to cover the language with a finite language. In particular, we show that a cover complexity measure is unbounded if it is induced by a certain class of context-free grammars with a bounded number of nonterminals on the right-hand side of their productions. Moreover, we obtain an analogous result for strict regular and strict linear grammars. Thirdly, in Section 5, we show—for these restricted kinds of context-free grammars—that we can reduce the cover complexity of a finite language L to the minimum of the exact complexities over a finite number of supersets L' of L. In Section 6, we construct a regular cover-incompressible sequence of finite languages. Finally, in Section 7, relying on, among other results, the cover-incompressible sequence constructed in Section 6, we investigate the grammatical cover complexity of the language operations intersection, union, and concatenation on finite languages for context-free, (strict) linear, and (strict) regular grammars.

This paper extends [25] in the following respects: we construct a cover-incompressible sequence of finite languages that generalises the one constructed in [20,21]. Based on this more general cover-incompressible sequence, we prove a lower bound on the cover complexity of union w.r.t. a fixed alphabet and, moreover, we include fixes of some flaws as well as full proofs of many of the results that have been stated without proof in [25].

2. Cover complexity

In this section, we introduce the basic definitions of the notion of cover complexity from both an abstract and grammatical point of view. Moreover, in order to fix notation and terminology, we also introduce the basic notions of formal language theory that are relevant to this paper.

For a set A, we write $\mathcal{P}_{\text{fin}}(A)$ for the set of finite subsets of A. Let Σ be an alphabet, then a function $\mu : \mathcal{P}_{\text{fin}}(\Sigma^*) \to \mathbb{N}$ is called Σ-*complexity measure*. If the alphabet is irrelevant or clear from the context, we will just speak about a complexity measure. Let μ be a Σ-complexity measure, then the *cover complexity measure induced by* μ is the Σ-complexity measure $\mu\mathsf{c}$ defined as

$$\mu\mathsf{c}(L) = \min\{\, \mu(L') \mid L \subseteq L' \in \mathcal{P}_{\text{fin}}(\Sigma^*) \,\}.$$

Note that the minimum is well-defined even though there are infinitely many $L' \in \mathcal{P}_{\text{fin}}(\Sigma^*)$ with $L \subseteq L'$, since μ maps to the natural numbers. We have $\mu\mathsf{c}(L) \leq \mu(L)$, for all $L \in \mathcal{P}_{\text{fin}}(\Sigma^*)$, and, moreover, for every $L \in \mathcal{P}_{\text{fin}}(\Sigma^*)$, there is an $L' \supseteq L$ such that $\mu\mathsf{c}(L) = \mu(L')$. A Σ-complexity measure μ is called *bounded* if there is a $k \in \mathbb{N}$ such that $\mu(L) \leq k$, for all $L \in \mathcal{P}_{\text{fin}}(\Sigma^*)$, and unbounded otherwise.

A *context-free* (CF) grammar is a quadruple $G = (N, \Sigma, P, S)$, where N and Σ are disjoint finite sets of *nonterminals* and *terminals*, respectively, $S \in N$ is the *start symbol*, and P is a finite set of *productions* of the form $A \to \alpha$, where $A \in N$ and $\alpha \in (N \cup \Sigma)^*$. Let A be a nonterminal, then a production with A on its left-hand side is called A-*production*. We write P_A for the subset of A-productions in P, i.e., $P_A = \{\, A \to \alpha \mid A \to \alpha \in P \,\}$ and, for $N' \subseteq N$, we define $P_{N'} = \bigcup_{A \in N'} P_A$. A production of the form $S \to w$, for $w \in \Sigma^*$, is called *trivial*; all other productions are called *non-trivial*. Let $G = (N, \Sigma, P, S)$, then we define $G_{\text{t}} = (N, \Sigma, P_{\text{t}}, S)$, where P_{t} is the set of trivial productions of G. If $G = G_{\text{t}}$, then G is called *trivial grammar* and *non-trivial grammar* otherwise. The set of all words of length at most k, for $k \geq 0$, over Σ is denoted by $\Sigma^{\leq k}$. We also consider further restrictions of context-free grammars: a context-free grammar is called *linear context-free* (LIN) if all productions in G are of the form $A \to \alpha$, where $\alpha \in \Sigma^*(N \cup \{\varepsilon\})\Sigma^*$; a context-free grammar is called *right-linear* or *regular* (REG) if all productions in G are of the form $A \to \alpha$, where $\alpha \in \Sigma^*(N \cup \{\varepsilon\})$. Moreover, a context-free grammar is called *strict linear* (SLIN) if all productions are of the form $A \to aBb$ or $A \to c$, where $B \in N$ and $a, b, c \in \Sigma^{\leq 1}$. Similarly, a context-free grammar is called *strict regular* (SREG) if all productions are of the form $A \to aB$ or $A \to b$, where $B \in N$ and $a, b \in \Sigma^{\leq 1}$. We will also write SREG, REG, ... for the set of strict regular, regular, ... grammars and set $\Gamma = \{\text{SREG}, \text{REG}, \text{SLIN}, \text{LIN}, \text{CF}\}$ and $\Gamma_s = \{\text{SREG}, \text{SLIN}\}$. As usual, the *derivation relation of* G is denoted by $\Rightarrow_G$ and the reflexive and transitive closure of $\Rightarrow_G$ is written as $\Rightarrow_G^*$. If the grammar is clear from the context, we will often omit the subscript G. For a nonterminal A of G, the *language of* A *w.r.t.* G is defined as $L_A(G) = \{\, w \in \Sigma^* \mid A \Rightarrow_G^* w \,\}$. The *language of a grammar* G is then defined as $L(G) = L_S(G)$. We say that a context-free grammar G covers a language L if $L(G) \supseteq L$. In our setting, the *size* of a context-free grammar $G = (N, \Sigma, P, S)$ is defined as $|G| = |P|$. We say that a word $v \in \Sigma^*$ is a *subword* of a word $w \in \Sigma^*$ if there are words $v_1, v_2 \in \Sigma^*$ such that $w = v_1 v v_2$ and, moreover, we say that a word $v \in \Sigma^*$ is a *prefix* of a word $w \in \Sigma^*$ if there is some word $u \in \Sigma^*$ such that $w = vu$. Let $L \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*)$ and $X \in \Gamma$, then the X-complexity of L is defined as

$$\mathsf{Xc}(L) = \min\{\, |G| \mid G \in X, L = L(G) \,\}.$$

Clearly, Xc is a complexity measure and induces the cover complexity measure

$$\mathsf{Xcc}(L) = \min\{\, \mathsf{Xc}(L') \mid L \subseteq L' \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*) \,\}.$$

Consequently, we say that G is a *minimal* X-grammar *covering* (or *generating*, respectively) the finite language L if $L(G)$ is *finite*, $L \subseteq L(G)$ (or $L = L(G)$, respectively), and $|G| = \mathsf{Xcc}(L)$ (or $|G| = \mathsf{Xc}(L)$, respectively). Note that, in general, there may be more than one minimal X-grammar for a given finite language L. The following result shows the existence of regular cover-incompressible sequences of finite languages and has been proved in [20,21].

Theorem 1. *For all $n \geq 1$, there is a language L_n with $|L_n| = n = \mathsf{REGcc}(L_n)$.*

On the other hand, for every finite language L, there is a trivial context-free grammar covering L with a constant number of productions:

Theorem 2. *Let $L \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*)$, then $\mathsf{CFcc}(L) \leq |\Sigma| + 2$.*

Proof. Let $\Sigma = \{a_1, a_2, \ldots, a_n\}$, $l = \max\{\, |w| \mid w \in L \,\}$, and consider the grammar G consisting of the productions $S \to A^l$, $A \to a_1 \mid a_2 \mid \cdots \mid a_n \mid \varepsilon$. Then $\mathrm{L}(G) = \Sigma^{\leq l} \supseteq L$. □

3. Unboundedness of cover complexity measures

Motivated by the above Theorems 1 and 2, in this section, we will characterise the situations in which a cover complexity measure collapses to a bounded complexity measure. Before we can give this characterisation, we need some auxiliary results on "almost inverting" functions from $\mathbb{N}$ to $\mathbb{N}$. These will be provided in Lemmas 3 and 4. A function $f : \mathbb{N} \to \mathbb{N}$ is called *bounded* if there is a $k \in \mathbb{N}$ such that $f(n) \leq k$, for all $n \in \mathbb{N}$, and *unbounded* otherwise. Moreover, a function f is called *monotonic* if $n \leq m$ implies $f(n) \leq f(m)$.

Lemma 3. *Let $f : \mathbb{N} \to \mathbb{N}$ be a monotonic and unbounded function and define $g : \mathbb{N} \to \mathbb{N}, n \mapsto \min\{\, i \in \mathbb{N} \mid n \leq f(i) \,\}$, then g is well-defined, monotonic, unbounded, and, for all $x, y \in \mathbb{N}$, we have $g(x) \leq y$ iff $x \leq f(y)$.*

Proof. The function g is well-defined because, due to unboundedness of f, there is at least one $i \in \mathbb{N}$ with $f(i) \geq n$.

Moreover, g is unbounded for suppose there is a $k \in \mathbb{N}$ such that $g(n) = \min\{i \in \mathbb{N} \mid n \leq f(i)\} \leq k$, for all $n \in \mathbb{N}$. Then, in particular by monotonicity of f,

$$g(f(k+1)) = \min\{\, i \in \mathbb{N} \mid f(k+1) \leq f(i) \,\} = k+1,$$

which contradicts $g(n) \leq k$, for all $n \in \mathbb{N}$.

Also, g is monotonic for if $n \leq m$, then for any $x \in \mathbb{N}$ with $m \leq f(x)$, we also have $n \leq f(x)$ and thus $g(n) \leq x$, in particular for $x = g(m)$.

Let $x, y \in \mathbb{N}$. If $x \leq f(y)$, then we have $g(x) = \min\{i \in \mathbb{N} \mid x \leq f(i)\} \leq y$. On the other hand, if $g(x) \leq y$, then we have both $f(g(x)) \leq f(y)$ and $f(g(x)) = f(\min\{i \in \mathbb{N} \mid x \leq f(i)\}) \geq x$. □

Lemma 4. *Let $g : \mathbb{N} \to \mathbb{N}$ be a monotonic and unbounded function and define $f : \mathbb{N} \to \mathbb{N}, n \mapsto \max\{\, i \in \mathbb{N} \mid g(i) \leq n \,\}$. Then f is well-defined, monotonic, unbounded, and, for all $x, y \in \mathbb{N}$, we have $g(x) \leq y$ iff $x \leq f(y)$.*

Proof. The function f is well-defined because, for each $n \in \mathbb{N}$, there are only finitely many $i \in \mathbb{N}$ with $g(i) \leq n$, for suppose there would be infinitely many such i, then, by monotonicity, $g(j) \leq n$, for all j after a certain $j_0 \in \mathbb{N}$. This, however, contradicts the unboundedness of g.

Moreover, f is unbounded for suppose there is a $k \in \mathbb{N}$ such that, for all $n \in \mathbb{N}$, we have $f(n) = \max\{\, i \in \mathbb{N} \mid g(i) \leq n \,\} \leq k$, then, in particular by monotonicity of g,

$$f(g(k+1)) = \max\{\, i \in \mathbb{N} \mid g(i) \leq g(k+1) \,\} = k+1,$$

which contradicts $f(n) \leq k$, for all $n \in \mathbb{N}$.

Also, f is monotonic for if $n \leq m$, then for any $x \in \mathbb{N}$ with $g(x) \leq n$, we have $g(x) \leq m$ and thus $f(m) \geq x$, in particular for $x = f(n)$.

Let $x, y \in \mathbb{N}$. If $g(x) \leq y$, then we have $f(y) = \max\{i \in \mathbb{N} \mid g(i) \leq y\} \geq x$. If $x \leq f(y)$, then we have $g(x) \leq g(f(y)) = g(\max\{\, i \in \mathbb{N} \mid g(i) \leq y \,\}) \leq y$. □

Examples for “almost inverting” functions on the natural numbers are the exponential function 2^n and the function that applies the ceiling function to the binary logarithm of n:

Example 1. Let $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$ be functions defined as follows:

$$f(n) = 2^n \qquad \text{and} \qquad g(n) = \begin{cases} 0 & \text{if } n = 0, \\ \lceil \log n \rceil & \text{if } n > 0. \end{cases}$$

Clearly, both f and g are monotonic and unbounded functions that, in addition, satisfy

$$f(n) = \max\{\, i \in \mathbb{N} \mid g(i) \leq n \,\} \quad \text{and} \quad g(n) = \min\{\, i \in \mathbb{N} \mid n \leq f(i) \,\}.$$

Thus, by Lemmas 3 and 4, it holds that

$$g(x) \leq y \text{ iff } x \leq f(y),$$

for all $x, y \in \mathbb{N}$.

A complexity measure $\rho : \mathcal{P}_{\text{fin}}(\Sigma^*) \to \mathbb{N}$ is called *reference complexity measure* if ρ is unbounded and $L_1 \subseteq L_2$ implies $\rho(L_1) \leq \rho(L_2)$. For reference complexity measures, what we have in mind are, e.g., the number of words $|L|$ in a language or their cumulated lengths $\|L\| = \sum_{w \in L} |w|$. Let μ be a complexity measure, then a reference complexity measure ρ is called *reference complexity measure for* μ if $\mu(L) \leq \rho(L)$, for all finite languages L. Typical examples for the above definitions include: $\mu \in \{\mathsf{REGc}, \mathsf{LINc}, \mathsf{CFc}\}$ and $\rho(L) = |L|$, or μ is the minimal size, that is, symbolic complexity of a regular, linear, or context-free grammar and $\rho(L) = \|L\|$. In the following theorem, a characterisation of the unboundedness of a cover complexity measure $\mu\mathsf{c}$ in terms of the existence of a relation between μ and a reference complexity measure ρ for μ is provided.

Theorem 5. *Let μ be an unbounded Σ-complexity measure and ρ be a reference complexity measure for μ, then the following conditions are equivalent:*

1. *$\mu\mathsf{c}$ is unbounded.*
2. *There is a monotonic and unbounded function $f : \mathbb{N} \to \mathbb{N}$ such that, for all $L \in \mathcal{P}_{\text{fin}}(\Sigma^*)$, we have $\rho(L) \leq f(\mu(L))$.*
3. *There is a monotonic and unbounded function $g : \mathbb{N} \to \mathbb{N}$ such that, for all $L \in \mathcal{P}_{\text{fin}}(\Sigma^*)$, we have $g(\rho(L)) \leq \mu(L)$.*

Proof. 2. ⇒ 3. has been shown in Lemma 3, and 3. ⇒ 2. in Lemma 4.

For 3. ⇒ 1., let $L \in \mathcal{P}_{\text{fin}}(\Sigma^*)$, then, by definition of $\mu\mathsf{c}$, there is some $L' \in \mathcal{P}_{\text{fin}}(\Sigma^*)$ such that $L \subseteq L'$ and $\mu\mathsf{c}(L) = \mu(L')$. Therefore,

$$\mu\mathsf{c}(L) = \mu(L') \geq g(\rho(L')) \geq g(\rho(L)),$$

where the first inequality follows from the assumption of condition 3., and the second one follows by definition of ρ and from the fact that g is a monotonic function. This shows the unboundedness of $\mu\mathsf{c}$ based on the unboundedness of g and ρ.

For showing 1. ⇒ 3., we argue by contraposition. Assume that every function $g : \mathbb{N} \to \mathbb{N}$ such that $g(\rho(L)) \leq \mu(L)$ for all $L \in \mathcal{P}_{\text{fin}}(\Sigma^*)$ is bounded or not monotonic. Consider $h : \mathbb{N} \to \mathbb{N}, n \mapsto \min\{\, \mu(L) \mid \rho(L) \geq n, L \in \mathcal{P}_{\text{fin}}(\Sigma^*) \,\}$ and note that, due to the unboundedness of ρ, h is well-defined. Moreover, we have $h(\rho(L)) \leq \mu(L)$. Now, we prove that h is monotonic and therefore, by assumption, also bounded. For monotonicity, let $n \leq m$. Then we have

$$\{\, L \in \mathcal{P}_{\text{fin}}(\Sigma^*) \mid \rho(L) \geq m \geq n \,\} \subseteq \{\, L \in \mathcal{P}_{\text{fin}}(\Sigma^*) \mid \rho(L) \geq n \,\}$$

and therefore,

$$h(n) = \min\{\, \mu(L) \mid \rho(L) \geq n, L \in \mathcal{P}_{\text{fin}}(\Sigma^*) \,\} \leq \min\{\, \mu(L) \mid \rho(L) \geq m, L \in \mathcal{P}_{\text{fin}}(\Sigma^*) \,\} = h(m).$$

Thus, h is bounded, i.e., there is a $k \in \mathbb{N}$ and $(L_n)_{n \in \mathbb{N}}$ such that $n \mapsto \rho(L_n)$ is unbounded, but $\mu(L_n) \leq k$, for all $n \in \mathbb{N}$. Since, by definition of $\mu\mathsf{c}$,

$$\mu\mathsf{c}(L_n) \leq \mu(L_n) \leq k,$$

it follows that $\mu\mathsf{c}$ is bounded too. □

The subsequent theorem states that the cover complexity of a finite language L can be obtained from the minimum over the exact complexities of finite supersets of L whose reference complexity is bounded by a certain constant.

Theorem 6. *Let μ be a complexity measure and ρ be a reference complexity measure for μ. Then, for every finite language L, there is some $b \in \mathbb{N}$ such that*

$$\mu\mathsf{c}(L) = \min\{\, \mu(L') \mid L \subseteq L' \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*) \text{ and } \rho(L') \leq b \,\}.$$

Proof. If $\mu\mathsf{c}$ is bounded by k, let $b = k$. If $\mu\mathsf{c}$ is unbounded, then, by Theorem 5, there is a monotonic and unbounded function $g : \mathbb{N} \to \mathbb{N}$ such that $g(\rho(K)) \leq \mu(K)$, for all finite languages K, and, by Lemma 4, there is a monotonic and unbounded function $f : \mathbb{N} \to \mathbb{N}$ such that $g(x) > y$ iff $x > f(y)$, for all $x, y \in \mathbb{N}$. Let $b = f(\rho(L))$ and $L'' \supseteq L$ with $\rho(L'') > f(\rho(L))$, then $g(\rho(L'')) > \rho(L)$, and, since we have $\mu(L'') \geq g(\rho(L''))$, we obtain $\mu(L'') > \rho(L)$. Moreover, since $\rho(L) \geq \mu(L) \geq \mu\mathsf{c}(L)$, it follows that $\mu(L'') > \mu\mathsf{c}(L)$. □

The above theorem expresses $\mu\mathsf{c}$ in terms of μ and ρ. Depending on ρ, the set of covers L' of L that is used to determine $\mu\mathsf{c}(L)$ may or may not be a finite set. We will analyse the reduction of $\mu\mathsf{c}(L)$ to the value of $\mu(\cdot)$ on a finite set more thoroughly in Section 5.

4. Unboundedness of grammatical cover complexity measures

After dealing with complexity measures in an abstract sense in the previous section, we will now come back to applications in the realm of context-free grammars. In particular, we will now apply Theorem 5 to the number of productions in various types of context-free grammars. Hence, we will fix $\rho(L) = |L|$ as reference complexity measure.

The subsequent lemma was already shown in [6] and implies that Xcc, for $X \in \{\mathsf{SREG}, \mathsf{REG}, \mathsf{SLIN}, \mathsf{LIN}\}$, is an unbounded complexity measure.

Lemma 7 *([6], Lemma 2.3). Let G be a linear grammar with n productions generating a finite language, then* $|\mathrm{L}(G)| \leq 2^{n-1}$.

Corollary 8. *The measures* SREGcc, REGcc, SLINcc, *and* LINcc *are unbounded.*

Proof. Define the function $f : \mathbb{N} \to \mathbb{N}, n \mapsto 2^n$. Clearly, f is both monotonic and unbounded. By Lemma 7, for all finite languages $L \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*)$, we have

$$\rho(L) = |L| \leq 2^{\mathsf{LINc}(L)-1} \leq 2^{\mathsf{LINc}(L)} = f(\mathsf{LINc}(L))$$

and hence, by Theorem 5, LINcc is unbounded. The unboundedness of the measures SREGcc, REGcc, and SLINcc follows from the facts that

$$\mathsf{LINcc}(L) \leq \mathsf{SLINcc}(L) \leq \mathsf{SREGcc}(L) \quad \text{and} \quad \mathsf{LINcc}(L) \leq \mathsf{REGcc}(L),$$

for all finite languages $L \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*)$. □

The following definition of *class of* CFGs in terms of closure under identifying nonterminals, omission of productions, and containment of all trivial grammars is motivated by the subsequent proofs of Lemmas 10 and 13.

Definition 1. A set X of context-free grammars is called *class of context-free grammars* if

1. $(N, \Sigma, P, S) \in \mathsf{X}$ and $p \in P$ implies $(N, \Sigma, P \setminus \{p\}, S) \in \mathsf{X}$,
2. X is closed under identifying two nonterminals, and
3. for each finite language L, X contains the trivial grammar generating L.

We now show that any context-free grammar can be transformed into a grammar, where every nonterminal derives at least one non-empty word without increasing the number of productions. Grammars of this kind are said to be in *pruned normal form*.

Definition 2. A context-free grammar $G = (N, \Sigma, P, S)$ is in *pruned normal form* (PNF) if, for all nonterminals $A \in N \setminus \{S\}$, we have that $L_A(G) \nsubseteq \{\varepsilon\}$, and there are $\alpha_1, \alpha_2 \in (N \cup \Sigma)^*$ and a word $w \in L(G)$ such that $S \Rightarrow_G^* \alpha_1 A \alpha_2 \Rightarrow_G^* w$.

Note that if $L_A(G) = \emptyset$, then there is no A-production that derives a string that solely consists of terminal symbols, i.e., no A-production contributes to the derivation of a word in $L(G)$. By definition, a grammar in pruned normal form does not contain such useless nonterminals.

Lemma 9. *Let G be a context-free grammar. Then there is a context-free grammar G' in* PNF *with $|G'| \leq |G|$ and $L(G') = L(G)$.*

Proof. Let $G = (N, \Sigma, P, S)$ be a context-free grammar. If G is already in PNF, let $G' = G$; so assume that G is not in PNF. In the case that $L(G) = \emptyset$, let $G' = (\{S\}, \Sigma, \emptyset, S)$, and if $L(G) = \{\varepsilon\}$, let $G' = (\{S\}, \Sigma, \{S \to \varepsilon\}, S)$. In both of these cases, G' is a context-free grammar in PNF with $|G'| \leq |G|$ and $L(G') = L(G)$.

Thus, assume $L(G) \not\subseteq \{\varepsilon\}$ and that there is some $A \in N \setminus \{S\}$ such that A does not occur in any derivation of a word in $L(G)$. We construct a grammar G' from G by removing the nonterminal A and all A-productions (i.e., productions in which A occurs on the left-hand side) as well as all productions in which A occurs on the right-hand side. We repeat this step until we have constructed a grammar $G'' = (N'', \Sigma, P'', S)$ such that for all $A \in N'' \setminus \{S\}$, there are $\alpha_1, \alpha_2 \in (N'' \cup \Sigma)^*$ and a word $w \in L(G'')$ with

$$S \Rightarrow^*_{G''} \alpha_1 A \alpha_2 \Rightarrow^*_{G''} w.$$

Then it follows that $L_A(G) \neq \emptyset$, for all $A \in N''$. It is also easy to see that $L(G'') = L(G)$ and $|G''| \leq |G|$. In the case that there is some nonterminal $A \in N'' \setminus \{S\}$ with $L_A(G'') = \{\varepsilon\}$, we construct a context-free grammar G''' by omitting the nonterminal A and all A-productions, and replacing all occurrences of A on the right-hand sides of productions in G'' by ε. Then, clearly, $L(G''') = L(G'') = L(G)$ and $|G'''| \leq |G''| \leq |G|$. We repeat this step until we have constructed a grammar G^* which contains no nonterminal $A \neq S$ with $L_A(G^*) = \{\varepsilon\}$. This transformation of G into G^* clearly terminates as in each step the number of nonterminals decreases. □

In order to illustrate the construction steps carried out in the proof of Lemma 9, we give the following example of transforming a context-free grammar into pruned normal form.

Example 2. Let $G = (N, \Sigma, P, S)$ be a context-free grammar with the following set of productions P:

$$\begin{aligned} S &\to aA_1 \mid bA_1 \mid B \mid aBC, \\ A_1 &\to aA_2 \mid bA_2 \mid B, \\ A_2 &\to a \mid b \mid B, \\ B &\to \varepsilon, \\ C &\to D. \end{aligned}$$

Clearly, $L(G) = \{a, b\}^{\leq 3} \not\subseteq \{\varepsilon\}$, and observe that $L_B(G) = \{\varepsilon\}$ and $L_C(G) = L_D(G) = \emptyset$. We follow the proof of Lemma 9 in order to construct a context-free grammar in PNF that is equivalent to G. Thus, we omit all B- and C-productions as well as the production $S \to aBC$ and replace each occurrence of B by ε in the remaining productions. This yields a grammar $G' = (\{S, A_1, A_2\}, \Sigma, P', S)$ with the following set of productions P':

$$\begin{aligned} S &\to aA_1 \mid bA_1 \mid \varepsilon, \\ A_1 &\to aA_2 \mid bA_2 \mid \varepsilon, \\ A_2 &\to a \mid b \mid \varepsilon. \end{aligned}$$

We clearly have that $L(G') = L(G) = \{a, b\}^{\leq 3}$ and $|G'| \leq |G|$.

A context-free grammar $G = (N, \Sigma, P, S)$ is called *cyclic*[1] if there is some nonterminal $A \in N$ such that $A \Rightarrow^+_G \alpha_1 A \alpha_2$, for $\alpha_1, \alpha_2 \in (N \cup \Sigma)^*$; otherwise G is called *acyclic*. We now show that if a grammar G belongs to a class of context-free grammars, then that class also contains an equivalent acyclic grammar with at most $|G|$ productions.

Lemma 10. *Let* X *be a class of* CFG*s or* $\mathsf{X} \in \Gamma_S$*. If* $G \in \mathsf{X}$ *and* $L(G)$ *is finite, then there is an acyclic* $G' \in \mathsf{X}$ *with* $|G'| \leq |G|$ *and* $L(G') = L(G)$*.*

Proof. If G is acyclic, then define $G' = G$. Therefore, assume that G is cyclic, i.e., there is some $A_1 \in N$ and $\beta_1, \beta_2 \in (N \cup \Sigma)^*$ such that $A_1 \Rightarrow^+_G \beta_1 A_1 \beta_2$. By Lemma 9, we can assume, without loss of generality, that G is in PNF, i.e., for all $B \in N \setminus \{S\}$, we have $L_B(G) \not\subseteq \{\varepsilon\}$, and B is reachable from S and used to derive a word w in $L(G)$: there are $\alpha_1, \alpha_2 \in (N \cup \Sigma)^*$ such that $S \Rightarrow^*_G \alpha_1 B \alpha_2 \Rightarrow^*_G w$. If $\beta_1\beta_2 \neq \varepsilon$, then, since A_1 is reachable from S and we have $A_1 \Rightarrow^+_G \beta_1 A_1 \beta_2$, we can derive infinitely many words, i.e., $L(G)$ is infinite. Contradiction. If $\beta_1\beta_2 = \varepsilon$, then there is a derivation of the form

$$A_1 \Rightarrow_G A_2 \Rightarrow_G \ldots \Rightarrow_G A_n \Rightarrow_G A_1,$$

[1] Note that the definition of a *cyclic* grammar slightly differs from that of a *self-embedding* one: a grammar $G = (N, \Sigma, P, S)$ is called *self-embedding* if there is some nonterminal $A \in N$ such that $A \Rightarrow^*_G \alpha_1 A \alpha_2$, for $\alpha_1, \alpha_2 \in \Sigma^+$; otherwise G is called *non self-embedding*.

for $A_1, A_2, \dots, A_n \in N$ and $n \geq 1$. Note that the case that $A_1 \Rightarrow_G^* \gamma \Rightarrow_G^* A_1$, for $\gamma \in N^{\geq 2}$, is impossible for otherwise, due to G being in PNF, we could derive infinitely many words. We define a new grammar G^* from G with $|G^*| \leq |G|$ by identifying the nonterminal $A_1, A_2, \dots, A_n$ with a nonterminal $A \notin N$. That is, we replace all A_i in G, for $1 \leq i \leq n$, by A. Thus, every G-derivation can be transformed into a G^*-derivation, and, vice versa, every G^*-derivation can be transformed into a G-derivation by adding suitable productions of the form $A_i \to A_j$. Consequently, we have $L(G^*) = L(G)$. Note that G^* still contains a production of the form $A \to A$. Let G' be the grammar obtained from removing the production $A \to A$ from G^*. Then $G' \in \mathsf{X}$, $|G'| < |G^*| \leq |G|$, and $L(G') = L(G)$. □

The following result shows that Lemma 7 can be generalised from linear to context-free grammars which contain only a bounded number of nonterminals on the right-hand side of each of their productions:

Lemma 11. *Let G be a grammar with n productions generating a finite language such that every production of G contains at most k nonterminals on its right-hand side. Then* $|\mathrm{L}(G)| \leq n^{(k+1)^n}$.

Proof. We proceed by induction on the number of nonterminals p in G and show that $|L(G)| \leq n^{(k+1)^p}$. In light of Lemma 10, we can assume that G is acyclic.

For the base case, assume that $p = 1$, i.e., the grammar G contains a single nonterminal S. Since G is acyclic, S cannot occur on the right-hand side of any production. Thus, $k = 0$ and $L(G)$ contains exactly $n = n^{(0+1)^1}$ words.

For the induction step, assume that G consists of n productions and contains the nonterminals $A_1, A_2, \dots, A_{p+1}$ such that every production with left-hand side A_i only contains nonterminals A_j with $i > j$. We can assume the latter, since, by acyclicity of G, we can fix a linear order on the nonterminals in the above sense. The nonterminal A_1 is clearly minimal, i.e., cannot contain any nonterminals on the right-hand side of its productions. Thus, the productions with left-hand side A_1 are of the form $A_1 \to w_1 \mid w_2 \mid \dots \mid w_m$ with $w_i \in \Sigma^*$, for $1 \leq i \leq m \leq n$. Moreover, let $B \to \alpha$ be an arbitrary production of G with $B \neq A_1$. We define the grammar G' from G by replacing $B \to \alpha$ by the productions $B \to \alpha_1 \mid \alpha_2 \mid \dots \mid \alpha_{m'}$ such that the α_i, for $1 \leq i \leq m'$, are all possible combinations of replacing the occurrences of the nonterminal A_1 in α by the words $w_1, w_2, \dots, w_m$. Clearly, $m' \leq m^k \leq n^k$. Moreover, we remove the nonterminal A_1 together with all A_1-productions. Since this step is repeated for all non-minimal nonterminals, the grammar G' contains at most $n \cdot n^k = n^{k+1}$ productions and p nonterminals. Furthermore, we have $L(G') = L(G)$. By induction hypothesis, we get that

$$|L(G')| = |L(G)| \leq (n^{k+1})^{(k+1)^p} = n^{(k+1)^{p+1}}.$$

Since in such a grammar G, there are at most n nonterminals, we immediately get $|L(G)| \leq n^{(k+1)^n}$. □

Corollary 12. *Let* X *be a class of* CFGs *with a bounded number of nonterminals occurring on the right-hand side of each production. Then* Xcc *is unbounded.*

Proof. Let $G \in \mathsf{X}$ contain n production rules and let k be the bound on the number of nonterminals occurring on the right-hand side of each production. Define $f : \mathbb{N} \to \mathbb{N}, n \mapsto n^{(k+1)^n}$. Clearly, f is both monotonic and unbounded. By Lemma 11, for all finite languages $L \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*)$, we have

$$\rho(L) = |L| \leq \mathsf{Xc}(L)^{(k+1)^{\mathsf{Xc}(L)}} = f(\mathsf{Xc}(L)).$$

Hence, by Theorem 5, Xcc is unbounded. □

A context-free grammar $G = (N, \Sigma, P, S)$ is said to be in *Chomsky normal form* if all productions are of the form $A \to BC$, $A \to a$, or $S \to \varepsilon$, where $A, B, C \in N$ and $a \in \Sigma$.

Example 3. An immediate consequence of Corollary 12 is that for the class CNF of grammars in Chomsky normal form, CNFcc is an unbounded complexity measure. Moreover, by Lemma 11, the number of words generated by a grammar G in CNF with n productions is bounded above by n^{3^n}, i.e., $|L(G)| \leq n^{3^n}$.

5. Computing cover complexity from exact complexity

We now turn to characterising the cover complexity of a finite language L based on the exact complexity of a finite set of finite languages related to L. To this end, we first show the following simple lemmas.

Lemma 13. *Let* X *be a class of* CFGs, *L be a finite language,* $\ell := \max\{\, |w| \mid w \in L \,\}$, *and G be a minimal* X*-grammar with* $L(G) \supseteq L$. *Then for all productions of the form* $A \to u_0 B_1 u_1 B_2 \cdots B_n u_n$ *of G with* $u_0, u_1, \dots, u_n \in \Sigma^*$, *we have* $|u_0 u_1 \cdots u_n| \leq \ell$.

Proof. Let $G = (N, \Sigma, P, S)$ and suppose that there is a production $p\colon A \to u_0 B_1 u_1 B_2 \cdots B_n u_n \in P$ with $|u_0 \cdots u_n| > \ell$. Then there is no G-derivation of a word in L that uses p, and so the X-grammar $G' = (N, \Sigma, P \setminus \{p\}, S)$ satisfies both $L(G') \supseteq L$ and $|G'| < |G|$. Contradiction to the minimality of G. □

Any linear grammar that covers a language whose longest word has length ℓ can generate only words of length at most the number of words in L times ℓ.

Lemma 14. *Let L be a finite language, $\ell := \max\{\,|w| \mid w \in L\,\}$, and G be a minimal* LIN*-grammar with $L(G) \supseteq L$. Then $\max\{\,|w| \mid w \in L(G)\,\} \le |L| \cdot \ell$.*

Proof. In light of Lemma 10, we can assume that G is acyclic. Let $w \in L(G)$ be arbitrary. Then there is a derivation δ of w in G of the form

$$S \Rightarrow_G \alpha_1 \Rightarrow_G \alpha_2 \Rightarrow_G \ldots \Rightarrow_G \alpha_n = w,$$

where $\alpha_i \in (N \cup \Sigma)^*$, for $1 \le i \le n$. Due to the fact that G is acyclic, each production of G can occur at most once in a G-derivation. Thus, by Lemma 13, it follows that each derivation step in δ can add at most ℓ letters to the previous intermediate string, i.e., $|w| \le n * \ell$. Since $n \le |G| \le |L|$, we get $|w| \le |L| * \ell$. □

Since, in general, it does not hold that $|G| \le |L|$ if G is a minimal X-grammar covering L, for $X \in \Gamma_s$, we get a different bound on the length of a longest word in strict regular and strict linear grammars.

Lemma 15. *Let $X \in \Gamma_s$ and $L \subseteq \Sigma^{\le \ell}$. Then $\mathsf{Xc}(L) \le 1 + \sum_{i=1}^{\ell} i \cdot |\Sigma|^i$.*

Proof. Consider the trivial grammar G generating $L = \{\,w_1, w_2, \ldots, w_n\,\}$, i.e., each production of G is of the form $S \to w_i = a_{i,1}a_{i,2}\ldots a_{i,k}$ with $a_{i,j} \in \Sigma \cup \{\varepsilon\}$, for $1 \le i \le n$ and $1 \le j \le k \le \ell$. We break up each trivial production $S \to w_i = a_{i,1}a_{i,2}\ldots a_{i,k}$ with $a_{i,j} \in \Sigma \cup \{\varepsilon\}$, for $1 \le i \le n$ and $1 \le j \le k \le \ell$, into the following strict regular productions:

$$\begin{aligned} S &\to a_{i,1}A_{i,2} \\ A_{i,2} &\to a_{i,2}A_{i,3} \\ &\vdots \\ A_{i,k-1} &\to a_{i,k-1}A_{i,k} \\ A_{i,k} &\to a_{i,k}, \end{aligned}$$

where, for each $i \in \{1, 2, \ldots, n\}$, the $A_{i,1}, A_{i,2}, \ldots, A_{i,k}$ are fresh nonterminals. Consequently, if we assume that $\ell = \max\{\,|w| \mid w \in L\,\}$, we get that we need at most $k \cdot |\Sigma|^k$, for each $k \in \{1, 2, \ldots, \ell\}$, strict regular productions, since there are $|\Sigma|^k$ many words of length k. This amounts to

$$\mathsf{SREGc}(L) \le 1 + \sum_{i=1}^{\ell} i \cdot |\Sigma|^i,$$

for every finite language $L \subseteq \Sigma^{\le \ell}$. The result for strict linear grammars follows immediately, since $\mathsf{SLINc}(L) \le \mathsf{SREGc}(L)$, for all finite languages L. □

Lemma 16. *Let $\ell \ge 0$, $L \subseteq \Sigma^{\le \ell}$ be a finite language, and G be a minimal X-grammar, for $X \in \Gamma_s$, with $L(G) \supseteq L$. Then $\max\{\,|w| \mid w \in L(G)\,\} \le \ell + \ell^3 \cdot |\Sigma|^\ell$.*

Proof. The proof is essentially the same as the proof of Lemma 14, but instead of $n \le |G| \le |L|$, we have $n \le |G| \le 1 + \sum_{i=1}^{\ell} i \cdot |\Sigma|^i \le 1 + \ell^2 \cdot |\Sigma|^\ell$ by Lemma 15, since one can show by induction on ℓ that $\sum_{i=1}^{\ell} i \cdot |\Sigma|^i \le \ell^2 \cdot |\Sigma|^\ell$. □

In the case of a context-free grammar G that covers a finite language L where the number of nonterminals occurring on the right-hand side of each production in G is bounded by some $k \ge 2$, we get that any such grammar can produce only words of length at most ℓ times $k^{|L|}$.

Lemma 17. *Let* X *be a class of* CFGs *such that every production in an* X*-grammar contains at most $k \ge 2$ nonterminals on its right-hand side, let L be a finite language, $\ell := \max\{\,|w| \mid w \in L\,\}$, and G be a minimal* X*-grammar with $L(G) \supseteq L$. Then $\max\{\,|w| \mid w \in L(G)\,\} \le \ell \cdot k^{|L|}$.*

Proof. In light of Lemma 10, we can assume that G is acyclic and therefore fix a linear order on the nonterminals $A_1, A_2, \dots, A_p$ in the following sense: every production with left-hand side A_i only contains nonterminals A_j with $j < i$. We show, by induction on q, that every derivation that starts with an A_j with $j \leq q \leq p$ has at most $\sum_{i=0}^{q-1} k^i$ steps. If $q = 1$, then a derivation has at most one step. On the other hand, if $q > 1$, then the first step replaces A_j with at most k occurrences of nonterminals which are some A_h with $h \leq q-1$. By induction hypothesis, each of them has a derivation of length at most $\sum_{i=0}^{q-2} k^i$ and there are at most k of them, so the total number of steps in the derivation is at most $1 + k\sum_{i=0}^{q-2} k^i = \sum_{i=0}^{q-1} k^i$. Moreover, for $k \geq 2$, we have $\sum_{i=0}^{q-1} k^i \leq k^q$, since $k-1 \geq 1$ implies $k^q \leq k^q \cdot (k-1)$. As a consequence, $\frac{k^q}{k-1} \leq k^q$ and thus $\sum_{i=0}^{q-1} k^i = \frac{k^q-1}{k-1} \leq k^q$. The result is then obtained from $q \leq p \leq |G| \leq |L|$. Similarly as in the proof of Lemma 14, from acyclicity of G and Lemma 13, it follows that any word $w \in L(G)$ has length at most $\ell \cdot k^{|L|}$. □

What the following theorem tells us is that, for a certain class of context-free grammars, we can obtain the cover complexity of a given finite language L in terms of the minimum over the exact complexities of a finite number of finite covers L' of L.

Theorem 18. *Let* X *be a class of* CFGs *such that every production in an* X*-grammar contains at most k nonterminals on its right-hand side. Then, for every finite language L, there is a finite set $\mathcal{S}_L$ of finite languages such that*

$$\mathsf{Xcc}(L) = \min\{\, \mathsf{Xc}(L') \mid L' \in \mathcal{S}_L \,\}.$$

Proof. Let G be an arbitrary minimal X-grammar with n productions covering a finite language L, i.e., $\mathsf{Xcc}(L) = n$, and let $\ell = \max\{\, |w| \mid w \in L \,\}$. Clearly, $n \leq |L|$. We distinguish two cases. In the case that $k = 1$, G is a linear grammar and so according to Lemmas 7 and 14, every X-grammar covering L is an X-grammar generating a finite language $L' \supseteq L$ that satisfies both

$$\mathsf{Xc}(L') \leq |L'| \leq 2^{|L|-1} \quad \text{and} \quad \max\{\, |w| \mid w \in L' \,\} \leq \ell \cdot |L|.$$

Since, by definition, $\mathsf{Xcc}(L) = \min\{\, \mathsf{Xc}(L') \mid L \subseteq L' \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*) \,\}$, setting

$$\mathcal{S}_{L,1} = \{\, L' \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*) \mid L \subseteq L', |L'| \leq 2^{|L|-1}, \max\{|w| \mid w \in L'\} \leq \ell \cdot |L| \,\}$$

yields the conclusion that $\mathsf{Xcc}(L) = \min\{\, \mathsf{Xc}(L') \mid L' \in \mathcal{S}_{L,1} \,\}$. Similarly, in the case that $k \geq 2$, the conclusion $\mathsf{Xcc}(L) = \min\{\, \mathsf{Xc}(L') \mid L' \in \mathcal{S}_{L,k} \,\}$ follows from Lemmas 11 and 17 by setting

$$\mathcal{S}_{L,k} = \{\, L' \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*) \mid L \subseteq L', |L'| \leq |L|^{(k+1)^{|L|}}, \max\{|w| \mid w \in L'\} \leq \ell \cdot k^{|L|} \,\}.$$

Clearly, each of the sets $\mathcal{S}_{L,k}$, for $k \geq 1$, satisfies the conditions of Theorem 18. □

Both the set of strict regular and the set of strict linear grammars are not classes of context-free grammars in the sense of Definition 1, as both of these sets do not contain all trivial grammars. Therefore, we have to adapt the proof strategy in order to arrive at a result for strict regular and strict linear grammars that is analogous to Theorem 18.

Theorem 19. *Let $X \in \Gamma_s$. Then, for every finite language L, there is a finite set $\mathcal{S}_L$ of finite languages such that*

$$\mathsf{Xcc}(L) = \min\{\, \mathsf{Xc}(L') \mid L' \in \mathcal{S}_L \,\}.$$

Proof. Let G be an arbitrary minimal X-grammar with n productions covering a finite language L, i.e., $\mathsf{Xcc}(L) = n$, and let $\ell = \max\{\, |w| \mid w \in L \,\}$. By Lemma 15 and the fact that $\mathsf{SLINcc}(L) \leq \mathsf{SREGcc}(L)$, for all finite languages L, we have

$$n \leq 1 + \sum_{i=1}^{\ell} i \cdot |\Sigma|^i \leq 1 + \ell^2 + |\Sigma|^\ell.$$

According to Lemmas 7 and 16, every X-grammar covering L is an X-grammar generating a finite language $L' \supseteq L$ that satisfies both $|L'| \leq 2^{\mathsf{Xc}(L')-1} = 2^{\mathsf{Xcc}(L)-1} \leq 2^{\ell^2+|\Sigma|^\ell}$ and $\max\{\, |w| \mid w \in L' \,\} \leq \ell + \ell^3 + |\Sigma|^\ell$. Therefore, by setting

$$\mathcal{S}_L = \{\, L' \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*) \mid L \subseteq L', |L'| \leq 2^{\ell^2+|\Sigma|^\ell}, \max\{\, |w| \mid w \in L'\} \leq \ell + \ell^3 \cdot |\Sigma|^\ell \,\},$$

the conclusion follows. Clearly, the set $\mathcal{S}_L$ satisfies the condition of Theorem 19. □

So, for a class of CFGs as in Theorem 18 as well as for strict regular and strict linear grammars, determining the cover complexity of L boils down to computing the exact complexity on the finite set $\mathcal{S}_L$.

6. A cover-incompressible sequence of languages

In this section, we are going to construct a regular cover-incompressible sequence of finite languages. This sequence is similar to, yet more general than, the one defined in [20,21]. The need for a more general sequence is motivated by the fact that it allows us to show that the bound on the regular cover complexity of union is tight w.r.t. a fixed alphabet (see Section 7.2). This new sequence consists of so-called *segmented languages*, i.e., languages in which all words are repetitions of a *separator symbol* followed by a so-called *building block*. More formally, this is defined as follows:

Definition 3. Let Σ be an alphabet not containing the letter s. Then we write Σ_s for $\Sigma \cup \{s\}$. A word $w \in \Sigma_s^*$ such that $w = (sv)^k$, for some $k \geq 1$ and some $v \in \Sigma^+$, is called *segmented word*. The word v and the letter s are called the *building block* and the separator symbol, respectively, of w. Occurrences of v in w are called segments. A segmented word $(sv)^k$ with $|v| = \ell$ is called a (k,ℓ)*- segmented word*. A language consisting of (k,ℓ)-segmented words only is called a (k,ℓ)*-segmented language*.

A finite language L is called X cover-incompressible, for $X \in \{\mathsf{REG}, \mathsf{LIN}, \mathsf{CF}\}$, if any X-grammar covering L contains at least as many productions as there are words in L. The notion of cover (in-)compressibility can also be extended to sequences of finite languages.

Definition 4. Let L be a finite language. Then L is called X *cover-compressible*, for $X \in \{\mathsf{REG}, \mathsf{LIN}, \mathsf{CF}\}$, if $\mathsf{Xcc}(L) < |L|$ and X *cover-incompressible* otherwise.

A sequence $(L_n)_{n\geq 1}$ of finite languages is called X *cover-incompressible*, for $X \in \{\mathsf{REG}, \mathsf{LIN}, \mathsf{CF}\}$, if there is an $M \in \mathbb{N}$ such that for all $n \geq M$, the language L_n is X cover-incompressible. A sequence $(L_n)_{n\geq 1}$ of finite languages is called X *cover-compressible* if for every $M \in \mathbb{N}$, there is an $n \geq M$ such that L_n is X cover-compressible.

Note that it is trivial to construct a cover-incompressible sequence of languages of constant size, e.g., $L_n = \{a\}$, for a letter a. It is also trivial to construct a sequence of cover-incompressible languages in an infinite alphabet, e.g., $L_n = \{a_1, a_2, \ldots, a_n\}$, for letters $a_1, a_2, \ldots$. Consequently, in this section, we will construct a regular cover-incompressible sequence of languages of unbounded size over a finite alphabet:

Let Σ be an arbitrary alphabet not containing the letter s. For all $n \geq 1$, let $a_n \in \mathbb{N}$, let $\ell, k \colon \mathbb{N} \to \mathbb{N}$, and let $A_n \subseteq \Sigma^*$ such that

$$\begin{aligned} &\ell(n) \leq \lceil \log(a_n) \rceil\,, \\ &k(n) \geq \left\lceil \frac{9 \cdot a_n}{\ell(n)+1} \right\rceil, \text{ and} \\ &A_n \subseteq \Sigma^{\ell(n)} \text{ with } |A_n| = a_n. \end{aligned}$$

Then, for each $n \geq 1$, we write $[\ell(n), k(n), A_n]$ for the language

$$\{\, (sw)^{k(n)} \mid w \in A_n \,\}.$$

Note that, for every $n \geq 1$, we have $|[\ell(n), k(n), A_n]| = |A_n| = a_n$ and all words in $[\ell(n), k(n), A_n]$ have the same length $k(n) \cdot (\ell(n)+1)$, i.e., $[\ell(n), k(n), A_n]$ is a $(k(n), \ell(n))$-segmented language for all $n \geq 1$. The number of segments has been chosen such that $k(n) \cdot (\ell(n)+1)$ is $9 \cdot a_n$ padded up to the next multiple of $\ell(n)+1$.

The above cover-incompressible sequence was obtained from the one constructed in [20,21] by relaxing the constraints on $\ell(n)$ and $k(n)$ from "=" to "≤" and "≥", respectively, and allowing arbitrary words of length $\ell(n)$ as building blocks for the segmented languages in the sequence. In the subsequent example, we demonstrate how we have to choose the parameters in order to obtain the cover-incompressible sequence constructed in [20,21] from the above more general sequence.

Example 4. For $n \geq 1$ and $k \in \{0, 1, \ldots, 2^n - 1\}$, we write $b_n(k) \in \{0,1\}^n$ for the n-bit binary representation of k. Let, for all $n \geq 1$,

$$\begin{aligned} &a_n = n, \\ &\ell(n) = \lceil \log(a_n) \rceil\,, \text{ and} \\ &k(n) = \left\lceil \frac{9 \cdot a_n}{\ell(n)+1} \right\rceil, \\ &A_n = \{\, b_{\ell(n)}(i) \mid 0 \leq i \leq n-1 \,\}. \end{aligned}$$

	$\mathsf{Xcc}(L_1 \cap L_2)$	$\mathsf{Xcc}(L_1 \cup L_2)$	$\mathsf{Xcc}(L_1 L_2)$
LIN	$\mathbf{min\{c_1, c_2\}}$	$c_1 + c_2$	$\min\{d_1 + c_2, c_1 + d_2\}$
SLIN	$\mathbf{min\{c_1, c_2\}}$	$\mathbf{c_1 + c_2}$	$\min\{d_1 + c_2, c_1 + d_2\}$
REG	$\mathbf{min\{c_1, c_2\}}$	$\mathbf{c_1 + c_2}$	$c_1 + c_2$
SREG	$\mathbf{min\{c_1, c_2\}}$	$\mathbf{c_1 + c_2}$	$\mathbf{c_1 + c_2}$

Fig. 1. Summary of results. For $i \in \{1, 2\}$, let $c_i = \mathsf{Xcc}(L_i)$ and $d_i = \mathsf{(S)REGcc}(L_i)$.

Note that we have both $|A_n| = a_n = n$ and $A_n \subseteq \{0, 1\}^{\ell(n)}$. As a consequence,

$$[\ell(n), k(n), A_n] = \{ (sw)^{k(n)} \mid w \in A_n \} = \{ (sb_{\ell(n)}(i))^{k(n)} \mid 0 \leq i \leq n - 1 \},$$

which is equal to the language L_n constructed in [21, Definition 14].

We note that a slight modification of the proof of [21, Theorem 1] leads to the following cover-incompressibility result.

Theorem 20. *Any sequence* $([\ell(n), k(n), A_n])_{n \geq 1}$ *is* REG *cover-incompressible.*

Proof Sketch. The proof is essentially the same as the one of [21, Theorem 1] for the less general regular cover-incompressible sequence. We just have to substitute the two inequalities $k(n) = \left\lceil \frac{9n}{\lceil \log(n) \rceil + 1} \right\rceil \geq \frac{9n}{\log(n)+2}$ and $n \geq |L'_n|$ by

$$k(n) \geq \left\lceil \frac{9 \cdot a_n}{\ell(n) + 1} \right\rceil \geq \left\lceil \frac{9 \cdot a_n}{\lceil \log(a_n) \rceil + 1} \right\rceil = \left\lceil \frac{9 \cdot |A_n|}{\lceil \log(|A_n|) \rceil + 1} \right\rceil \geq \frac{9 \cdot |A_n|}{\log(|A_n|) + 2}$$

and $|A_n| \geq |L'_n|$, respectively. The remaining parts of the proof are exactly as in the proof of [21, Theorem 1]. □

7. Bounds on language operations

In this section, we will prove upper and lower bounds on the cover complexity of the operations *intersection*, *union*, and *concatenation* of finite languages. While we have not yet been able to obtain matching lower bounds on union and concatenation w.r.t. fixed alphabets for all grammar types under consideration, we have been able to do so w.r.t. growing alphabets. The results of this section are summarised in Fig. 1, where **bold font** means that we have matching upper and lower bounds w.r.t. a fixed alphabet and non-bold means that the bounds are matching w.r.t. a growing alphabet. For the remainder of this section, let $\Delta = \Gamma \setminus \{\mathsf{CF}\}$.

7.1. Intersection

The bound on the cover complexity of intersecting two finite languages L_1 and L_2 corresponds to the minimum of the cover complexities of L_1 and L_2, and this bound is tight as shown in Theorem 22.

Theorem 21. *Let* $X \in \Delta$ *and* L_1 *and* L_2 *be finite languages. Then*

$$\mathsf{Xcc}(L_1 \cap L_2) \leq \min\{ \mathsf{Xcc}(L_1), \mathsf{Xcc}(L_2) \}.$$

Proof. Let G_i be a minimal X-grammar with $L(G_i) \supseteq L_i$, for $i \in \{1, 2\}$; then $L(G_i) \supseteq L_1 \cap L_2$. Simply choose $G = G_i$ with $|G_i| = \min\{ |G_1|, |G_2| \}$. □

In order to show that the bound of Theorem 21 is tight, we can use the fact that Xcc, for $X \in \Delta$, is an unbounded complexity measure (see Corollary 8).

Theorem 22. *Let* $X \in \Delta$. *Then there exists a finite alphabet* Σ *such that for all* $n_1, n_2 \geq 1$, *there are finite languages* L_1 *and* L_2 *with* $\mathsf{Xcc}(L_1) \geq n_1$ *and* $\mathsf{Xcc}(L_2) \geq n_2$ *such that*

$$\mathsf{Xcc}(L_1 \cap L_2) \geq \min\{ \mathsf{Xcc}(L_1), \mathsf{Xcc}(L_2) \}.$$

Proof. Let Σ be an arbitrary finite alphabet and let $n_1, n_2 \geq 1$ such that, without loss of generality, $n_2 \geq n_1$. From Corollary 8, it follows that there are languages $L_1, L_2 \in \mathcal{P}_{\text{fin}}(\Sigma^*)$ with $\mathsf{Xcc}(L_1) \geq n_1$ and $\mathsf{Xcc}(L_2) \geq n_2$. Define $L'_2 = L_1 \cup L_2$. Then $\mathsf{Xcc}(L'_2) \geq \mathsf{Xcc}(L_1) \geq n_1$, for otherwise there would be a grammar covering $L'_2 \supseteq L_1$ with less than $\mathsf{Xcc}(L_1)$ productions. A similar argument shows that $\mathsf{Xcc}(L'_2) \geq \mathsf{Xcc}(L_2) \geq n_2$. Thus, we clearly have

$$\mathsf{Xcc}(L_1 \cap L'_2) = \mathsf{Xcc}(L_1) = \min\{ \mathsf{Xcc}(L_1), \mathsf{Xcc}(L'_2) \}.$$

This proves the stated claim. □

7.2. Union

The bound on the cover complexity of the union of two finite languages L_1 and L_2 corresponds to the sum of the cover complexities of L_1 and L_2, and, for strict regular, regular, and strict linear grammars, this bound is tight w.r.t. a fixed alphabet as shown in Theorems 26 and 27.

Theorem 23. *Let $X \in \Delta$ and L_1 and L_2 be finite languages. Then*

$$\mathsf{Xcc}(L_1 \cup L_2) \leq \mathsf{Xcc}(L_1) + \mathsf{Xcc}(L_2).$$

Proof. Let $X \in \Delta$ and, for $i \in \{1,2\}$, $G_i = (N_i, \Sigma_i, P_i, S_i)$ be a minimal X-grammar with $L(G_i) \supseteq L_i$ and $|G_i| = \mathsf{Xcc}(L_i)$ such that $N_1 \cap N_2 = \emptyset$. By minimality of G_i and since, by Lemma 10, we can also assume that it is acyclic, S_i does not occur on the right-hand side of a production in P_i. Let $S \notin N_1 \cup N_2$; we define $G = (N_1 \cup N_2 \cup S, \Sigma_1 \cup \Sigma_2, P, S)$, where

$$P = \{ S \to \alpha \mid S_1 \to \alpha \in P_1 \text{ or } S_2 \to \alpha \in P_2 \} \cup \{ A \to \alpha \in P_1 \mid A \neq S_1 \} \cup \{ A \to \alpha \in P_2 \mid A \neq S_2 \}.$$

Clearly, we have $L(G) = L(G_1) \cup L(G_2) \supseteq L_1 \cup L_2$ and $|G| = |G_1| + |G_2|$, that is, $\mathsf{Xcc}(L_1 \cup L_2) \leq \mathsf{Xcc}(L_1) + \mathsf{Xcc}(L_2)$. Moreover, $G_1, G_2 \in X$ implies $G \in X$. □

If we consider growing alphabets, then we can show that the above upper bound on the cover complexity of union is tight for all considered grammar types.

Theorem 24. *Let $X \in \Delta$. Then, for all $n_1, n_2 \geq 1$, there exists a finite alphabet Σ and finite languages L_1 and L_2 with $\mathsf{Xcc}(L_1) = n_1$ and $\mathsf{Xcc}(L_2) = n_2$ such that*

$$\mathsf{Xcc}(L_1 \cup L_2) \geq \mathsf{Xcc}(L_1) + \mathsf{Xcc}(L_2).$$

Proof. Let $n_1, n_2 \geq 1$. Then define $\Sigma = \{a_1, a_2, \ldots, a_{n_1}, b_1, b_2, \ldots, b_{n_2}\}$, $L_1 = \{a_1, a_2, \ldots, a_{n_1}\}$, and $L_2 = \{b_1, b_2, \ldots, b_{n_2}\}$. Thus, $L_1 \cup L_2 = \Sigma$. Moreover, we clearly have, $\mathsf{Xcc}(L_1) = n_1$, $\mathsf{Xcc}(L_2) = n_2$, and the language $L_1 \cup L_2$ can only be covered by a trivial grammar. Therefore,

$$\mathsf{Xcc}(L_1 \cup L_2) = n_1 + n_2 = \mathsf{Xcc}(L_1) + \mathsf{Xcc}(L_2).$$

This proves the stated claim. □

Now, we prove—w.r.t. a fixed alphabet—a lower bound on the strict linear cover complexity of union that matches the upper bound. To do so, we use the fact that in the case of strict regular and strict linear grammars, there is a connection between the number of productions and the length of a longest word in the generated finite language.

Lemma 25. *Let L be a finite language and $\ell = \max\{ |w| \mid w \in L \}$. Then*

$$\mathsf{SREGcc}(L) \geq \ell \quad \textit{and} \quad \mathsf{SLINcc}(L) \geq \left\lfloor \frac{\ell}{2} + 1 \right\rfloor.$$

Proof. Since the strict regular case can be shown using similar arguments, we only give a proof of the strict linear case. We will first show that in any minimal strict linear grammar $G = (N, \Sigma, P, S)$ the following statement holds:

$$\text{for all } A \in N \text{ and all } w \in \Sigma^*: \text{ if } A \Rightarrow_G^k w, \text{ then } k \geq \left\lfloor \frac{|w|}{2} + 1 \right\rfloor.$$

To prove the above statement, we will proceed by induction on the length of a derivation of w.

- **Base case:** Assume $k = 1$. If $A \Rightarrow_G w$, then, by definition of strict linear grammars, we must have that $w \in \Sigma \cup \{\varepsilon\}$, i.e., $|w| \leq 1$. Thus, we clearly have $k = 1 \geq \lfloor \frac{1}{2} + 1 \rfloor = \left\lfloor \frac{|w|}{2} + 1 \right\rfloor$.
- **Induction step:** Suppose $k \geq 2$ and $A \Rightarrow_G^k w$. We have to distinguish four cases according to the form of the derivation of w:
  1. Let $A \Rightarrow_G aBb \Rightarrow_G^{k-1} w = aw_1b$, for $a, b \in \Sigma$, $B \in N$, and $w_1 \in \Sigma^*$. Obviously, $B \Rightarrow_G^{k-1} w_1$. Thus, by induction hypothesis, it follows that $k - 1 \geq \left\lfloor \frac{|w_1|}{2} + 1 \right\rfloor$. This means

$$k \geq \left\lfloor \frac{|w_1|}{2} + 1 \right\rfloor + 1 = \left\lfloor \frac{|w_1|}{2} + 1 + 1 \right\rfloor = \left\lfloor \frac{|w_1|}{2} + \frac{2}{2} + 1 \right\rfloor = \left\lfloor \frac{|w_1| + 2}{2} + 1 \right\rfloor = \left\lfloor \frac{|w|}{2} + 1 \right\rfloor.$$

2. Next, consider $A \Rightarrow_G aB \Rightarrow_G^{k-1} w = aw_1$, for $a \in \Sigma$, $B \in N$, and $w_1 \in \Sigma^*$. The claim follows from similar arguments as in the first case.
3. Moreover, $A \Rightarrow_G Bb \Rightarrow_G^{k-1} w = w_1 b$, for $b \in \Sigma$, $B \in N$, and $w_1 \in \Sigma^*$. The claim follows from similar arguments as in the first case.
4. Finally, $A \Rightarrow_G B \Rightarrow_G^{k-1} w$, for $B \in N$ and $w \in \Sigma^*$. Obviously, we have $B \Rightarrow_G^{k-1} w$. By induction hypothesis, we get $k-1 \geq \left\lfloor \frac{|w|}{2} + 1 \right\rfloor$, which clearly implies that $k \geq \left\lfloor \frac{|w|}{2} + 1 \right\rfloor$.

Now, let L be a finite language over Σ with $\ell = \max\{|w| \mid w \in L\}$ and $G = (N, \Sigma, P, S)$ be a minimal strict linear grammar with $L(G) \supseteq L$. Then there is a derivation $\delta\colon S \Rightarrow_G^k w$ with $w \in \Sigma^\ell$ and $k \geq 1$. By the above statement, we get that $k \geq \left\lfloor \frac{|w|}{2} + 1 \right\rfloor = \left\lfloor \frac{\ell}{2} + 1 \right\rfloor$. By Lemma 10, we can assume, without loss of generality, that G is acyclic. Thus, since in an acyclic strict linear grammar all right-hand sides of productions contain at most one nonterminal, no production can occur twice in the derivation δ. As a consequence, the derivation δ uses k distinct productions in order to derive w. Hence,

$$\mathsf{SLINcc}(L) = |G| \geq k \geq \left\lfloor \frac{\ell}{2} + 1 \right\rfloor,$$

by minimality of G. □

Theorem 26. *There exists a finite alphabet Σ such that for all $n_1, n_2 \geq 1$, there are finite languages L_1 and L_2 with $\mathsf{SLINcc}(L_1) = n_1$ and $\mathsf{SLINcc}(L_2) = n_2$ such that*

$$\mathsf{SLINcc}(L_1 \cup L_2) \geq \mathsf{SLINcc}(L_1) + \mathsf{SLINcc}(L_2).$$

Proof. Let $\Sigma = \{a, b\}$ and, for $n_1, n_2 \geq 1$, we define the finite language $L = \{a^{2n_1-1}, b^{2n_2-1}\}$. Moreover, let $L_1 = \{a^{2n_1-1}\}$ and $L_2 = \{b^{2n_2-1}\}$. Clearly, we have $L = L_1 \cup L_2$ and from Lemma 25, we get that $\mathsf{SLINcc}(L_1) \geq \left\lfloor \frac{2n_1-1}{2} + 1 \right\rfloor = n_1$ and $\mathsf{SLINcc}(L_2) \geq \left\lfloor \frac{2n_2-1}{2} + 1 \right\rfloor = n_2$. It is easy to see that also $\mathsf{SLINcc}(L_1) \leq n_1$ and $\mathsf{SLINcc}(L_2) \leq n_2$. Since the languages L_1 and L_2 do not share a common letter, there can be no production that is used to derive words from both L_1 and L_2. Thus, we must have that

$$\mathsf{SLINcc}(L) = \mathsf{SLINcc}(L_1 \cup L_2) \geq \mathsf{SLINcc}(L_1) + \mathsf{SLINcc}(L_2).$$

This proves the stated claim. □

Finally, using segmented languages as defined in Section 6 and applying Theorem 20, we can show a lower bound on the (strict) regular cover complexity of union w.r.t. a fixed alphabet that matches the upper bound. At this point one may ask why the regular cover-incompressible sequence constructed in [21] is not suitable for the proof of the following lower bound. The simple answer is that the union of two sequences of this kind does not necessarily result in a sequence of this kind again. The more general cover-incompressible sequence of Section 6 allows, however, to define a cover-incompressible sequence of this kind that corresponds to the union of two such cover-incompressible sequences.

Theorem 27. *Let $X \in \{\mathsf{SREG}, \mathsf{REG}\}$. Then there exists an alphabet Σ such that for all $n_1, n_2 \geq 1$, there are finite languages L_1 and L_2 with $X\mathsf{cc}(L_1) \geq n_1$ and $X\mathsf{cc}(L_2) \geq n_2$ such that*

$$X\mathsf{cc}(L_1 \cup L_2) \geq X\mathsf{cc}(L_1) + X\mathsf{cc}(L_2).$$

Proof. Let $\Sigma_1 = \{a, b\}$, $\Sigma_2 = \{c, d\}$, and $\Sigma = \Sigma_1 \cup \Sigma_2$. Moreover, let $a_{n,1} = a_{n,2} = 2^{\lceil \log(n) \rceil}$ and $a_n = a_{n,1} + a_{n,2}$. We define two sequences of finite languages $(L_{1,n})_{n \geq 1}$ and $(L_{2,n})_{n \geq 1}$ with $L_{1,n} = [\ell(n), k(n), \Sigma_1^{\ell(n)}]$ and $L_{2,n} = [\ell(n), k(n), \Sigma_2^{\ell(n)}]$, for $n \geq 1$, based on:

$$\ell(n) := \lceil \log(n) \rceil = \left\lceil \log(a_{n,1}) \right\rceil = \left\lceil \log(a_{n,2}) \right\rceil \text{ and}$$
$$k(n) := \left\lceil \frac{9 \cdot a_n}{\ell(n) + 1} \right\rceil \geq \left\lceil \frac{9 \cdot a_{n,1}}{\ell(n) + 1} \right\rceil = \left\lceil \frac{9 \cdot a_{n,2}}{\ell(n) + 1} \right\rceil.$$

Recall that, for $i \in \{1, 2\}$,

$$L_{i,n} = [\ell(n), k(n), \Sigma_i^{\ell(n)}] = \{\, (sw)^{k(n)} \mid w \in \Sigma_i^{\ell(n)} \,\}.$$

Thus, clearly, $L_{i,n}$ is a $(k(n), \ell(n))$-segmented language, for $i \in \{1, 2\}$. Now, let us consider the sequence of finite languages $(L_n)_{n \geq 1}$ with

$$L_n = [\ell(n), k(n), \Sigma_1^{\ell(n)} \cup \Sigma_2^{\ell(n)}].$$

Clearly, $\ell(n) \leq \lceil \log(a_n) \rceil$ and, moreover,

$$L_n = [\ell(n), k(n), \Sigma_1^{\ell(n)} \cup \Sigma_2^{\ell(n)}] = \{ (sw)^{k(n)} \mid w \in \Sigma_1^{\ell(n)} \cup \Sigma_2^{\ell(n)} \}.$$

Thus, clearly, L_n is a $(k(n), \ell(n))$-segmented language. By Theorem 20, the sequences $(L_{1,n})_{n\geq 1}$, $(L_{2,n})_{n\geq 1}$, and $(L_n)_{n\geq 1}$ are REG cover-incompressible. Furthermore, we have that $L_{1,n} \cap L_{2,n} = \emptyset$ and $L_n = L_{1,n} \cup L_{2,n}$, for all $n \geq 1$.

By definition of REG cover-incompressibility, there is an M such that for all $n \geq M$, we have $\mathsf{REGcc}(L_{1,n}) = |L_{1,n}| \geq n$, $\mathsf{REGcc}(L_{2,n}) = |L_{2,n}| \geq n$, and $\mathsf{REGcc}(L_n) = |L_n| = |L_{1,n}| + |L_{2,n}| \geq n + n$. Let $m \geq M$ be such that $m \geq n_1, n_2$. Then $\mathsf{REGcc}(L_{1,m}) \geq m \geq n_1$ and $\mathsf{REGcc}(L_{2,m}) \geq m \geq n_2$. Consequently,

$$\mathsf{REGcc}(L_m) = \mathsf{REGcc}(L_{1,m} \cup L_{2,m}) = |L_m| = |L_{1,m}| + |L_{2,m}| = \mathsf{REGcc}(L_{1,m}) + \mathsf{REGcc}(L_{2,m}).$$

For the SREG-case, let $\Sigma = \{a, b\}$ and, for $n_1, n_2 \geq 1$, we define the finite languages $L_1 = \{a^{n_1}\}$ and $L_2 = \{b^{n_2}\}$. Moreover, let $L = L_1 \cup L_2$. Then, from Lemma 25, we get that $\mathsf{SREGcc}(L_1) \geq n_1$ and $\mathsf{SREGcc}(L_2) \geq n_2$. Since the words in L_1 and L_2 do not share a common letter, there can be no production that is used to derive words from both L_1 and L_2. Thus, we must have that

$$\mathsf{SREGcc}(L) = \mathsf{SREGcc}(L_1 \cup L_2) \geq \mathsf{SREGcc}(L_1) + \mathsf{SREGcc}(L_2).$$

This proves the stated claim. □

7.3. Concatenation

In contrast to the case of union, there is no uniform upper bound on the cover complexity of concatenating two finite languages for all grammar types under consideration. The reason for this is that the method used to combine two given regular grammars into a new regular grammar that covers the concatenation of their covered languages does not necessarily give us a linear grammar if we are given two linear grammars.

Theorem 28. *Let $X \in \{\mathsf{SREG}, \mathsf{REG}\}$ and L_1 and L_2 be finite languages. Then*

1. $\mathsf{Xcc}(L_1L_2) \leq \mathsf{Xcc}(L_1) + \mathsf{Xcc}(L_2)$,
2. $\mathsf{LINcc}(L_1L_2) \leq \min\{ \mathsf{REGcc}(L_1) + \mathsf{LINcc}(L_2), \mathsf{LINcc}(L_1) + \mathsf{REGcc}(L_2) \}$,
3. $\mathsf{SLINcc}(L_1L_2) \leq \min\{ \mathsf{SREGcc}(L_1) + \mathsf{SLINcc}(L_2), \mathsf{SLINcc}(L_1) + \mathsf{SREGcc}(L_2) \}$.

Proof. Let $G_i = (N_i, \Sigma_i, P_i, S_i)$ be a minimal X-grammar with $L(G_i) \supseteq L_i$ and $|G_i| = \mathsf{Xcc}(L_i)$, for $i \in \{1, 2\}$. Assume, w.l.o.g., $N_1 \cap N_2 = \emptyset$.

In order to show 1., let $X \in \{\mathsf{SREG}, \mathsf{REG}\}$ and define the grammar $G_{\mathsf{(S)REG}} = (N_1 \cup N_2, \Sigma_1 \cup \Sigma_2, P_{\mathsf{(S)REG}}, S_1)$, where

$$P_{\mathsf{(S)REG}} = \{ A \to wS_2 \mid A \to w \in P_1, w \in \Sigma^* \} \cup \{ A \to \alpha \mid A \to \alpha \in P_1, \alpha \notin \Sigma^* \} \cup P_2.$$

Note that in the strict regular case, the above construction of G_{SREG} preserves strict regularity because in a strict regular grammar the right-hand sides of productions without nonterminals are of length at most 1 and thus appending a single nonterminal to the right-hand side of such a production results again in a strict regular production. As a consequence, the above construction also works for strict regular grammars. Thus, $L(G_{\mathsf{(S)REG}}) = L(G_1)L(G_2) \supseteq L_1L_2$ and $|G_{\mathsf{(S)REG}}| = |G_1| + |G_2| = \mathsf{Xcc}(L_1) + \mathsf{Xcc}(L_2)$, which shows that

$$\mathsf{Xcc}(L_1L_2) \leq \mathsf{Xcc}(L_1) + \mathsf{Xcc}(L_2),$$

for all $L_1, L_2 \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*)$.

In order to show 2. and 3., let $G_{\mathsf{(S)REG},i} = (N_{\mathsf{(S)REG},i}, \Sigma_i, P_{\mathsf{(S)REG},i}, S_{\mathsf{(S)REG},i})$ and $G_{\mathsf{(S)LIN},i} = (N_{\mathsf{(S)LIN},i}, \Sigma_i, P_{\mathsf{(S)LIN},i}, S_{\mathsf{(S)LIN},i})$ be minimal (S)REG- and (S)LIN-grammars covering L_i, respectively, for $i \in \{1, 2\}$, that is, $L(G_{\mathsf{(S)REG},1}) \supseteq L_1$, $L(G_{\mathsf{(S)LIN},1}) \supseteq L_1$, $L(G_{\mathsf{(S)REG},2}) \supseteq L_2$, and $L(G_{\mathsf{(S)LIN},2}) \supseteq L_2$ as well as $|G_{\mathsf{(S)REG},i}| = \mathsf{(S)REGcc}(L_i)$ and $|G_{\mathsf{(S)LIN},i}| = \mathsf{(S)LINcc}(L_i)$.

It remains to show that there are (strict) linear grammars G_1 and G_2 such that $L(G_1) \supseteq L_1L_2$ and $L(G_2) \supseteq L_1L_2$ as well as $|G_1| \leq \mathsf{(S)REGcc}(L_1) + \mathsf{(S)LINcc}(L_2)$ and $|G_2| \leq \mathsf{(S)LINcc}(L_1) + \mathsf{(S)REGcc}(L_2)$. To this end, assume, without loss of generality, that $N_{\mathsf{(S)REG},i} \cap N_{\mathsf{(S)LIN},j} = \emptyset$, for $i \neq j$. Furthermore, we assume that $G_{\mathsf{(S)REG},1}$ is a right-linear and $G_{\mathsf{(S)REG},2}$ is a left-linear grammar. This assumption is needed for the following definition of the grammars G_1 and G_2, but it constitutes no restriction since every right-linear grammar can be transformed into a left-linear one (and vice versa) without increasing the number of productions. Next, we define two (strict) linear grammars

$$G_1 = (N_{\mathsf{(S)REG},1} \cup N_{\mathsf{(S)LIN},2}, \Sigma_1 \cup \Sigma_2, P_1, S_{\mathsf{(S)REG},1}) \text{ and}$$
$$G_2 = (N_{\mathsf{(S)LIN},1} \cup N_{\mathsf{(S)REG},2}, \Sigma_1 \cup \Sigma_2, P_2, S_{\mathsf{(S)REG},2}),$$

where

$$P_1 = \{\, A \to wS_{\mathsf{(S)LIN},2} \mid A \to w \in P_{\mathsf{(S)REG},1}, w \in \Sigma^* \,\} \cup \{\, A \to \alpha \in P_{\mathsf{(S)REG},1} \mid \alpha \notin \Sigma^* \,\} \cup P_{\mathsf{(S)LIN},2}$$

and

$$P_2 = \{\, A \to S_{\mathsf{(S)LIN},1} w \mid A \to w \in P_{\mathsf{(S)REG},2}, w \in \Sigma^* \,\} \cup \{\, A \to \alpha \in P_{\mathsf{(S)REG},2} \mid \alpha \notin \Sigma^* \,\} \cup P_{\mathsf{(S)LIN},1}.$$

Clearly, $|G_1| \le |G_{\mathsf{(S)REG},1}| + |G_{\mathsf{(S)LIN},2}|$ and $|G_2| \le |G_{\mathsf{(S)LIN},1}| + |G_{\mathsf{(S)REG},2}|$. Let $w_i \in L_i$, for $i \in \{1,2\}$, then $S_{\mathsf{(S)REG},i} \Rightarrow^* w_i$ and $S_{\mathsf{(S)LIN},i} \Rightarrow^* w_i$. Thus, by definition of G_1 and G_2, we get

$$S_{\mathsf{(S)REG},1} \Rightarrow^*_{G_1} w_1 S_{\mathsf{(S)LIN},2} \Rightarrow^*_{G_1} w_1 w_2$$

and

$$S_{\mathsf{(S)REG},2} \Rightarrow^*_{G_2} S_{\mathsf{(S)LIN},1} w_2 \Rightarrow^*_{G_2} w_1 w_2,$$

that is, $w_1 w_2 \in L(G_i)$, for $i \in \{1,2\}$. Therefore, $L(G_i) \supseteq L_1 L_2$, for $i \in \{1,2\}$. Thus, it follows that $|G_1| \le \mathsf{(S)REGcc}(L_1) + \mathsf{(S)LINcc}(L_2)$ and $|G_2| \le \mathsf{(S)LINcc}(L_1) + \mathsf{(S)REGcc}(L_2)$. Finally, since we have both $\mathsf{(S)LINcc}(L_1 L_2) \le |G_1|$ and $\mathsf{(S)LINcc}(L_1 L_2) \le |G_2|$, we choose the grammar with the fewest number of productions out of G_1 and G_2. This shows that both

$$\mathsf{LINcc}(L_1 L_2) \le \min\{\, \mathsf{REGcc}(L_1) + \mathsf{LINcc}(L_2), \mathsf{LINcc}(L_1) + \mathsf{REGcc}(L_2) \,\} \text{ and}$$
$$\mathsf{SLINcc}(L_1 L_2) \le \min\{\, \mathsf{SREGcc}(L_1) + \mathsf{SLINcc}(L_2), \mathsf{SLINcc}(L_1) + \mathsf{SREGcc}(L_2) \,\}$$

hold. □

The following lemma states some basic properties of minimal context-free grammars generating finite languages which are needed in the proof of Lemma 30.

Lemma 29 *([6], Lemma 2.1). Let $G = (N, \Sigma, P, S)$ be a minimal context-free grammar generating a finite language L. Then, for every $A \in N \setminus \{S\}$,*

1. *there are $\alpha_1, \alpha_2 \in (N \cup \Sigma)^*$ with $\alpha_1 \ne \alpha_2$ such that $A \to \alpha_1$ and $A \to \alpha_2$ are in P, and*
2. *the set $L_A(G) = \{\, w \in \Sigma^* \mid A \Rightarrow^*_G w \,\}$ contains at least two words.*

Now, we show that a grammar covering the concatenation of two disjoint alphabets (each containing at least two letters) needs at least as many productions as there are elements in their (disjoint) union. This lemma will play an important role in the proof of Theorem 31.

Lemma 30. *Let $\Sigma = \Sigma_1 \uplus \Sigma_2$ be a finite alphabet with $|\Sigma_1|, |\Sigma_2| \ge 2$. Then for all context-free grammars G with $L(G) \supseteq \Sigma_1 \Sigma_2$, we have $|G| \ge |\Sigma_1| + |\Sigma_2|$.*

Proof. In light of Lemma 10, we can assume, without loss of generality, that all grammars in this proof are acyclic. We proceed by induction on $|\Sigma|$.

- **Base case:** Let $|\Sigma| = 4$, i.e., $|\Sigma_1| = |\Sigma_2| = 2$, and assume, w.l.o.g., that $\Sigma = \{a_1, a_2\} \uplus \{b_1, b_2\}$. Then $\Sigma_1 \Sigma_2 = \{a_1 b_1, a_1 b_2, a_2 b_1, a_2 b_2\}$. Towards contradiction, assume that there is some context-free grammar $G = (N, \Sigma, P, S)$ with $L(G) \supseteq \Sigma_1 \Sigma_2$ and $|G| \le 3$. Clearly, G cannot be a trivial grammar, for otherwise G could not cover $\Sigma_1 \Sigma_2$. Thus, we can assume that G contains at least two distinct nonterminals S and A. By Lemma 29, it follows that there are productions $A \to v_1$ and $A \to v_2$ in P with $v_1 \ne v_2$, which means $|G| \ge 3$. Hence, P must be of the form

$$\{\, S \to \alpha, A \to v_1, A \to v_2 \,\},$$

where $\alpha \in (N \cup \Sigma)^*$ and $v_1, v_2 \in \Sigma^*$, as G is acyclic. We distinguish cases:
If $\alpha = A$, then $|L(G)| = 2$, i.e., G cannot cover $\Sigma_1 \Sigma_2$.
If $\alpha = A^n$, for $n \ge 2$, then we further distinguish the following cases:
1. If $v_1 = av'_1$ and $v_2 = a'v'_2$, for $a, a' \in \Sigma_1$ and $v'_1, v'_2 \in \Sigma^*$. In this case, we cannot derive words of length 2 ending with some $b \in \Sigma_2$ (even if both v'_1 and v'_2 do so).
2. If $v_1 = bv'_1$ and $v_2 = b'v'_2$, for $b, b' \in \Sigma_2$ and $v'_1, v'_2 \in \Sigma^*$. In this case, we can only derive words starting with b or b', but these kinds of words do not occur in $\Sigma_1 \Sigma_2$.
3. If $v_1 = av'_1$ and $v_2 = bv'_2$, for $a \in \Sigma_1$, $b \in \Sigma_2$, and $v'_1, v'_2 \in \Sigma^*$. In this case, we can only derive words starting with a fixed $a \in \Sigma_1$ or $b \in \Sigma_2$. As a consequence, we cannot derive words in $\Sigma_1 \Sigma_2$ that start with $a' \in \Sigma_1$, where $a \ne a'$.

4. If $v_1 = bv'_1$ and $v_2 = av'_2$, for $a \in \Sigma_1$, $b \in \Sigma_2$, and $v'_1, v'_2 \in \Sigma^*$. Analogous to the previous case.
5. If $v_1 = \varepsilon$ and $v_2 = cv'_2$, for $c \in \Sigma$ and $v'_2 \in \Sigma^*$. In this case, we can only derive words starting with a fixed $c \in \Sigma$. As a consequence, we cannot derive words in $\Sigma_1\Sigma_2$ that start with $a \in \Sigma_1$, where $a \neq c$.
6. If $v_1 = cv'_1$ and $v_2 = \varepsilon$, for $c \in \Sigma$ and $v'_1 \in \Sigma^*$. Analogous to the previous case.

If α has $w' \in \Sigma^+$ as subword, then G cannot derive all words occurring in $\Sigma_1\Sigma_2$, because there is no $w' \in \Sigma^+$ which is a subword of all $w \in \Sigma_1\Sigma_2$. Hence, we have $|G| \geq 4$.

- **Induction step:** Assume, without loss of generality, that $\Sigma = \Sigma_1 \uplus \Sigma_2$ with $\Sigma_2 = \Sigma'_2 \uplus \{b_{n+1}\}$, where $|\Sigma_1| = m$ and $|\Sigma_2| = n+1$. Towards contradiction, assume that there is some CF-grammar $G = (N, \Sigma, P, S)$ with $L(G) \supseteq \Sigma_1\Sigma_2$ and $|G| < m+n+1$.

  Define $L' = \Sigma_1\Sigma'_2$ and let $\Sigma(L') = \Sigma_1 \uplus \Sigma'_2$ denote the alphabet induced by the words in L'. Clearly, $|\Sigma(L')| = m+n$ and we can apply the induction hypothesis to obtain that for any CF-grammar G' with $L(G') \supseteq L'$, we have $|G'| \geq m+n$.

  Let $G'' = (N, \Sigma \setminus \{b_{n+1}\}, P'', S)$ be a CF-grammar obtained from G by defining

$$P'' = P \setminus \{\, A \to \alpha_1 b_{n+1} \alpha_2 \in P \mid \alpha_1, \alpha_2 \in (N \cup \Sigma)^* \,\}.$$

  Then it follows that $L(G'') \supseteq L'$, since $b_{n+1} \notin \Sigma(L')$.

  Note that any grammar covering $\Sigma_1\Sigma_2$ must contain at least one production whose right-hand side contains the letter b_{n+1}. Thus, $|G''| < m+n$, which contradicts the induction hypothesis. The case that $\Sigma = \Sigma_1 \uplus \Sigma_2$ with $\Sigma_1 = \Sigma'_1 \uplus \{a_{n+1}\}$, where $|\Sigma_1| = m+1$ and $\Sigma_2 = n$ can be shown using an analogous argument. □

If we consider growing alphabets, then we are able to show that the bounds of Theorem 28 are tight.

Theorem 31. *Let $X \in \{\mathsf{SREG}, \mathsf{REG}\}$. Then, for all $n_1, n_2 \geq 2$, there is a finite alphabet Σ and finite languages L_1 and L_2 with* $\mathsf{Xcc}(L_1) = n_1$ *and* $\mathsf{Xcc}(L_2) = n_2$ *such that*

1. $\mathsf{Xcc}(L_1L_2) \geq \mathsf{Xcc}(L_1) + \mathsf{Xcc}(L_2)$,
2. $\mathsf{LINcc}(L_1L_2) \geq \min\{\, \mathsf{REGcc}(L_1) + \mathsf{LINcc}(L_2), \mathsf{LINcc}(L_1) + \mathsf{REGcc}(L_2) \,\}$,
3. $\mathsf{SLINcc}(L_1L_2) \geq \min\{\, \mathsf{SREGcc}(L_1) + \mathsf{SLINcc}(L_2), \mathsf{SLINcc}(L_1) + \mathsf{SREGcc}(L_2) \,\}$.

Proof. Let $X \in \{\mathsf{SREG}, \mathsf{REG}\}$, $n_1, n_2 \geq 2$, and define the finite alphabet $\Sigma = \{a_1, a_2, \ldots, a_{n_1}, b_1, b_2, \ldots, b_{n_2}\}$ as well as the languages $L_1 = \{a_1, a_2, \ldots, a_{n_1}\}$ and $L_2 = \{b_1, b_2, \ldots, b_{n_2}\}$. Then, clearly, we have $\mathsf{Xcc}(L_1) = n_1$ and $\mathsf{Xcc}(L_2) = n_2$. Thus, since every X-grammar is context-free, we have, by Lemma 30, that

$$\mathsf{Xcc}(\Sigma) = \mathsf{Xcc}(L_1L_2) \geq n_1 + n_2 = \mathsf{Xcc}(L_1) + \mathsf{Xcc}(L_2)$$

as well as

$$\mathsf{SLINcc}(L_1L_2) \geq n_1 + n_2 = \min\{\, \mathsf{SREGcc}(L_1) + \mathsf{SLINcc}(L_2), \mathsf{SLINcc}(L_1) + \mathsf{SREGcc}(L_2) \,\}$$

and

$$\mathsf{LINcc}(L_1L_2) \geq n_1 + n_2 = \min\{\, \mathsf{REGcc}(L_1) + \mathsf{LINcc}(L_2), \mathsf{LINcc}(L_1) + \mathsf{REGcc}(L_2) \,\}.$$

This proves the stated claim. □

However, if we consider fixed alphabets, then, at this point, we are only able to show that the bound of Theorem 28 is tight for strict regular grammars.

Theorem 32. *There exists a finite alphabet Σ such that for all $n_1, n_2 \geq 1$, there exist finite languages L_1 and L_2 with* $\mathsf{SREGcc}(L_1) = n_1$ *and* $\mathsf{SREGcc}(L_2) = n_2$ *such that*

$$\mathsf{SREGcc}(L_1L_2) \geq \mathsf{SREGcc}(L_1) + \mathsf{SREGcc}(L_2).$$

Proof. Let $\Sigma = \{a\}$ and, for $n_1, n_2 \geq 1$, we define the finite language $L = \{a^{n_1+n_2}\}$. Moreover, let $L_1 = \{a^{n_1}\}$ and $L_2 = \{a^{n_2}\}$. Clearly, $L = L_1L_2$ and, from Lemma 25, we get that $\mathsf{SREGcc}(L_1) \geq n_1$ and $\mathsf{SREGcc}(L_2) \geq n_2$. It is easy to see that also $\mathsf{SREGcc}(L_1) \leq n_1$ and $\mathsf{SREGcc}(L_2) \leq n_2$. Again, by Lemma 25, it follows that

$$\mathsf{SREGcc}(L) = \mathsf{SREGcc}(L_1L_2) \geq n_1 + n_2 = \mathsf{SREGcc}(L_1) + \mathsf{SREGcc}(L_2).$$

This proves the stated claim. □

8. Conclusion

In this paper, we have investigated cover complexity measures for finite languages on three different levels of abstraction and shown that every complexity measure on finite languages naturally induces a corresponding cover complexity measure. We have characterised the situations in which arbitrary complexity measures thus obtained are unbounded. Based on these rather abstract results, we have shown that every class of context-free grammars that allows only a bounded number of nonterminals on the right-hand side of each production induces an unbounded production cover complexity measure. This, in turn, entails that the production cover complexity of a finite language L can be obtained as the minimum of the exact production complexities of a finite number of supersets L' of L. Moreover, we have investigated upper and lower bounds on the production cover complexity of the language operations intersection, union, and concatenation on finite languages for several different types of context-free grammars (see Fig. 1). In order to prove the tightness of the bounds on the regular cover complexity of union w.r.t. fixed alphabets, we have generalised the cover-incompressible sequence of languages constructed in [20,21] in a suitable fashion.

There is still a number of open problems w.r.t. the grammatical cover complexity of finite languages. At this point, we do not know whether the bounds on the regular and linear cover complexity of concatenation as well as the one on the linear cover complexity of union are tight w.r.t. a fixed alphabet. Moreover, the NP-completeness of the minimal cover problem for acyclic regular grammars without a fixed bound on the number of nonterminals is still open. In [24], the authors proved that it is in NP and conjectured that it is also NP-hard.

In summary, we believe that the study of the complexity of finite languages is a fruitful research area with strong ties to both proof theory and more classical questions of descriptional complexity.

Declaration of Competing Interest

There is no Competing Interest.

References

[1] J. Gruska, On a classification of context-free languages, Kybernetika 3 (1) (1967) 22–29.

[2] A. Cerný, Complexity and minimality of context-free grammars and languages, in: J. Gruska (Ed.), Mathematical Foundations of Computer Science (MFCS) 1977, in: Lecture Notes in Computer Science, vol. 53, Springer, 1977, pp. 263–271.

[3] J. Gruska, Some classifications of context-free languages, Inf. Control 14 (2) (1969) 152–179.

[4] J. Gruska, Complexity and unambiguity of context-free grammars and languages, Inf. Control 18 (5) (1971) 502–519.

[5] J. Gruska, On the size of context-free grammars, Kybernetika 8 (3) (1972) 213–218.

[6] W. Bucher, H.A. Maurer, K. Culik II, D. Wotschke, Concise description of finite languages, Theor. Comput. Sci. 14 (1981) 227–246.

[7] B. Alspach, P. Eades, G. Rose, A lower-bound for the number of productions required for a certain class of languages, Discrete Appl. Math. 6 (2) (1983) 109–115.

[8] W. Bucher, A note on a problem in the theory of grammatical complexity, Theor. Comput. Sci. 14 (1981) 337–344.

[9] W. Bucher, H.A. Maurer, K. Culik II, Context-free complexity of finite languages, Theor. Comput. Sci. 28 (1984) 277–285.

[10] J. Dassow, Descriptional complexity and operations—two non-classical cases, in: G. Pighizzini, C. Câmpeanu (Eds.), Descriptional Complexity of Formal Systems (DCFS), in: Lecture Notes in Computer Science, vol. 10316, Springer, Milano, Italy, 2017, pp. 33–44.

[11] J. Dassow, R. Harbich, Production complexity of some operations on context-free languages, in: M. Kutrib, N. Moreira, R. Reis (Eds.), Workshop on Descriptional Complexity of Formal Systems (DCFS), in: Lecture Notes in Computer Science, vol. 7386, Springer, Braga, Portugal, 2012, pp. 141–154.

[12] Z. Tuza, On the context-free production complexity of finite languages, Discrete Appl. Math. 18 (3) (1987) 293–304.

[13] C. Câmpeanu, N. Santean, S. Yu, Minimal cover-automata for finite languages, in: J. Champarnaud, D. Maurel, D. Ziadi (Eds.), International Workshop on Implementing Automata (WIA'98), in: Lecture Notes in Computer Science, vol. 1660, Springer, 1998, pp. 43–56.

[14] C. Câmpeanu, N. Santean, S. Yu, Minimal cover-automata for finite languages, Theor. Comput. Sci. 267 (1–2) (2001) 3–16.

[15] S. Hetzl, Applying tree languages in proof theory, in: A.-H. Dediu, C. Martín-Vide (Eds.), Language and Automata Theory and Applications, LATA, 2012, in: Lecture Notes in Computer Science, vol. 7183, Springer, 2012, pp. 301–312.

[16] S. Hetzl, A. Leitsch, G. Reis, J. Tapolczai, D. Weller, Introducing quantified cuts in logic with equality, in: S. Demri, D. Kapur, C. Weidenbach (Eds.), Automated Reasoning - 7th International Joint Conference, IJCAR, in: Lecture Notes in Computer Science, vol. 8562, Springer, 2014, pp. 240–254.

[17] S. Hetzl, A. Leitsch, G. Reis, D. Weller, Algorithmic introduction of quantified cuts, Theor. Comput. Sci. 549 (2014) 1–16.

[18] S. Hetzl, A. Leitsch, D. Weller, Towards algorithmic cut-introduction, in: Logic for Programming, Artificial Intelligence and Reasoning (LPAR-18), in: Lecture Notes in Computer Science, vol. 7180, Springer, 2012, pp. 228–242.

[19] G. Ebner, S. Hetzl, A. Leitsch, G. Reis, D. Weller, On the generation of quantified lemmas, J. Automat. Reason. 63 (1) (June 2019) 95–126, https://doi.org/10.1007/s10817-018-9462-8.

[20] S. Eberhard, S. Hetzl, Compressibility of finite languages by grammars, in: J. Shallit, A. Okhotin (Eds.), Descriptional Complexity of Formal Systems, DCFS, 2015, in: Lecture Notes in Computer Science, vol. 9118, Springer, 2015, pp. 93–104.

[21] S. Eberhard, S. Hetzl, On the compressibility of finite languages and formal proofs, Inf. Comput. 259 (2018) 191–213.

[22] P. Pudlák, Twelve problems in proof complexity, in: E.A. Hirsch, A.A. Razborov, A.L. Semenov, A. Slissenko (Eds.), Third International Computer Science Symposium in Russia (CSR), in: Lecture Notes in Computer Science, vol. 5010, Springer, 2008, pp. 13–27.

[23] M. Holzer, S. Wolfsteiner, On the grammatical complexity of finite languages, in: S. Konstantinidis, G. Pighizzini (Eds.), Descriptional Complexity of Formal Systems (DCFS), in: Lecture Notes in Computer Science, vol. 10952, Springer, Cham, 2018, pp. 151–162.

[24] S. Eberhard, G. Ebner, S. Hetzl, Complexity of decision problems on totally rigid acyclic tree grammars, in: M. Hoshi, S. Seki (Eds.), Developments in Language Theory (DLT), in: Lecture Notes in Computer Science, vol. 11088, Springer, Cham, 2018, pp. 291–303.

[25] S. Hetzl, S. Wolfsteiner, Cover complexity of finite languages, in: S. Konstantinidis, G. Pighizzini (Eds.), Descriptional Complexity of Formal Systems (DCFS), in: Lecture Notes in Computer Science, vol. 10952, Springer, Cham, 2018, pp. 139–150.