Cover Complexity of Finite Languages

Stefan Hetzl, Simon Wolfsteiner

Cover Complexity of Finite Languages*

Stefan Hetzl and Simon Wolfsteiner

TU Wien, Institute of Discrete Mathematics and Geometry,
Wiedner Hauptstraße 8–10, 1040 Wien, Austria
{stefan.hetzl,simon.wolfsteiner}@tuwien.ac.at

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.

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 [12], 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 [7, 13–15], where, among others, the number of productions of a grammar has been considered as complexity measure. In [4], 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., [1–3, 8, 9, 21].

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 [5, 6]: there, an automaton A is sought such that $L(A) \supseteq L$, but in addition it is required that $L(A) \setminus L$ 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 [16], there is an intimate relationship between a certain

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

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 [17–19]. In particular, [10, 11] 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 the central open problem in proof complexity [20]). The combinatorial center 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 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 context-free 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 class of context-free grammars with a bounded number of nonterminals on the right-hand side of their productions. Moreover, unboundedness allows to 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. Thirdly, and yet more specifically, in Section 5, 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.

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.

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. 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 $L \in \mathcal{P}_{\text{fin}}(\Sigma^*)$. 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*. 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 define $\Gamma = \{SREG, REG, SLIN, LIN, CF\}$. 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. The *language of a grammar* G is defined as $L(G) = \{\, w \in \Sigma^* \mid S \Rightarrow_G^* w \,\}$. We say that a context-free grammar G covers a language L if $L(G) \supseteq L$. The *size* of a context-free grammar $G = (N, \Sigma, P, S)$ is defined as $|G| = |P|$. Let $L \in \mathcal{P}_{\text{fin}}(\Sigma^*)$ and $X \in \Gamma$. Then the X-complexity of L is

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

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

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

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 language L. The following result shows the existence of regular-incompressible sequences of finite languages and has been proved in [10, 11].

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 $L \in \mathcal{P}_{\text{fin}}(\Sigma^*)$, 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 1 and 2. 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. The function f is called *monotone* if $n \leq m$ implies $f(n) \leq f(m)$.

Lemma 1. *Let $f : \mathbb{N} \to \mathbb{N}$ be monotone and unbounded, define $g : \mathbb{N} \to \mathbb{N}, n \mapsto \min\{\, i \in \mathbb{N} \mid n \leq f(i) \,\}$, then g is well-defined, monotone, unbounded, and, for all $x, y \in \mathbb{N}$: $g(x) \leq y$ iff $x \leq f(y)$.*

Lemma 2. *Let $g : \mathbb{N} \to \mathbb{N}$ be monotone and unbounded, let $f : \mathbb{N} \to \mathbb{N}, n \mapsto \max\{\, i \in \mathbb{N} \mid g(i) \leq n \,\}$. Then f is well-defined, monotone, unbounded, and, for all $x, y \in \mathbb{N}$: $g(x) \leq y$ iff $x \leq f(y)$.*

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 definition include: $\mu = \mathsf{REGc}, \mathsf{CFc}, \ldots$ and $\rho(L) = |L|$, or μ is the minimal size, that is, symbolic complexity of a regular, context-free, ... grammar and $\rho(L) = \|L\|$. The following theorem provides a characterisation of the unboundedness of a cover complexity measure.

Theorem 3. *Let μ be an unbounded Σ-complexity measure and ρ be a reference complexity measure for μ, then the following are equivalent:*

1. *$\mu\mathsf{c}$ is unbounded*
2. *there is a monotone and unbounded function $f : \mathbb{N} \to \mathbb{N}$ s.t. $\rho(L) \leq f(\mu(L))$, for all $L \in \mathcal{P}_{\text{fin}}(\Sigma^*)$.*
3. *there is a monotone and unbounded function $g : \mathbb{N} \to \mathbb{N}$ s.t. $g(\rho(L)) \leq \mu(L)$, for all $L \in \mathcal{P}_{\text{fin}}(\Sigma^*)$.*

Proof. 2. $\Rightarrow$ 3. has been shown in Lemma 1, and 3. $\Rightarrow$ 2. in Lemma 2.

For 3. $\Rightarrow$ 1., let $L \in \mathcal{P}_{\text{fin}}(\Sigma^*)$, then there is some $L' \in \mathcal{P}_{\text{fin}}(\Sigma^*)$ s.t. $L \subseteq L'$ and $\mu\mathsf{c}(L) = \mu(L')$. Therefore, $\mu\mathsf{c}(L) = \mu(L') \geq^{3.} g(\rho(L')) \geq^{\text{mon.}} g(\rho(L))$, which shows unboundedness of $\mu\mathsf{c}$ based on the unboundedness of g and ρ.

For showing 1. $\Rightarrow$ 3., we prove the contrapositive. Assume that every $g : \mathbb{N} \to \mathbb{N}$ s.t. $g(\rho(L)) \leq \mu(L)$, for all $L \in \mathcal{P}_{\text{fin}}(\Sigma^*)$, is bounded or not monotone. 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, $h(\rho(L)) \leq \mu(L)$. For monotonicity, let $n \leq m$. Then we have $\{\, L \in \mathcal{P}_{\text{fin}}(\Sigma^*) \mid \rho(L) \geq m \geq n \,\} \subseteq \{\, L \in \mathcal{P}_{\text{fin}}(\Sigma^*) \mid \rho(L) \geq n \,\}$. Therefore, $h(n) = \min\{\, \mu(L) \mid \rho(L) \geq$

$n\} \leq \min\{\, \mu(L) \mid \rho(L) \geq m\,\} = h(m)$. So 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 $\mu\mathsf{c}(L_n) \leq \mu(L_n) \leq k$, $\mu\mathsf{c}$ is bounded too. □

Theorem 4. *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 3, there is an unbounded and monotone function $g : \mathbb{N} \to \mathbb{N}$ s.t. $g(\rho(K)) \leq \mu(K)$, for all finite languages K, and, by Lemma 2, there is an unbounded and monotone 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 $\rho(L) \geq \mu(L) \geq \mu\mathsf{c}(L)$, we have $\mu(L'') > \mu\mathsf{c}(L)$. □

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 the next section.

4 Computing Cover Complexity from Exact Complexity

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

The subsequent lemma was already shown in [4] and implies that Xcc, for $X \in \{SREG, REG, SLIN, LIN\}$, is an unbounded complexity measure.

Lemma 3. *Let G be a linear grammar with n productions generating a finite language, then $|\mathrm{L}(G)| \leq 2^{n-1}$.*

Corollary 1. *The measures* SREGcc, REGcc, SLINcc, *and* LINcc *are unbounded.*

Proof. Define the function $f : \mathbb{N} \to \mathbb{N}, n \mapsto 2^{n-1}$. Clearly, f is both monotone and unbounded. By Lemma 3, for all finite languages $L \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*)$, we have $\rho(L) = |L| \leq 2^{\mathsf{LINc}(L)-1} = f(\mathsf{LINc}(L))$. Hence, by Theorem 3, LINcc is unbounded. The unboundedness of SREGcc, REGcc, and SLINcc follows from the fact that $\mathsf{LINcc}(L) \leq \mathsf{SLINcc}(L) \leq \mathsf{SREGcc}(L)$ and $\mathsf{LINcc}(L) \leq \mathsf{REGcc}(L)$, for all finite languages $L \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*)$. □

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 that $(N, \Sigma, P \setminus \{p\}, S) \in \mathsf{X}$ and 2. X is closed under identifying two nonterminals.*

A context-free grammar $G = (N, \Sigma, P, S)$ is called *self-embedding* if there is some $A \in N$ such that $A \Rightarrow_G^+ w_1 A w_2$, for $w_1, w_2 \in (N \cup \Sigma)^*$; otherwise G is called *non self-embedding*.

Lemma 4. *Let* X *be a class of context-free grammars. If* $G \in \mathsf{X}$ *and* $L(G)$ *is finite, then there is a non self-embedding* $G' \in \mathsf{X}$ *with* $|G'| \leq |G|$ *and* $L(G') = L(G)$.

The following result shows that Lemma 3 can be generalised from linear to context-free grammars that contain only a bounded number of nonterminals on the right-hand side of each of their productions:

Lemma 5. *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 Sketch. Since G generates a finite language, by Lemma 4, we can assume, without loss of generality, that it is non self-embedding. Thus, there is a nonterminal A whose productions are $A \to w_1 \mid w_2 \mid \ldots \mid w_m$ with $w_i \in \Sigma^*$, for $1 \leq i \leq m \leq n$. Replacing each occurrence of A by all of the w_i yields a grammar with less nonterminals. By iterating this operation, one obtains a trivial grammar with the above mentioned bound. □

Corollary 2. *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 monotone and unbounded. By Lemma 5, 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 3, Xcc is unbounded. □

An immediate consequence of Corollary 2 is that for the class CNF of grammars in Chomsky normal form[1], CNFcc is an unbounded complexity measure. Moreover, by Lemma 5, the number of words generated by a grammar G in CNF with n productions is bounded above by n^{3^n}, i.e., $|L(G)| \leq n^{3^n}$.

Now, we show that the right-hand side of each production in a minimal context-free grammar covering a language whose longest word has length ℓ contains at most ℓ terminals.

Lemma 6. *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* $u_0, u_1, \ldots, u_n \in \Sigma^*$*, we have* $|u_0 u_1 \cdots u_n| \leq \ell$.

[1] 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 $A \to \varepsilon$, where $A, B, C \in N$ and $a \in \Sigma$.

Lemma 7. *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 $\max\{\,|w| \mid w \in L(G)\,\} \leq |L| \cdot \ell$.*

Lemma 8. *Let X be a class of CFGs such that every production in an X-grammar contains at most $k \geq 2$ 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 $\max\{\,|w| \mid w \in L(G)\,\} \leq \ell \cdot k^{|L|}$.*

Proof Sketch. Since G generates a finite language, by Lemma 4, we can assume, without loss of generality, that it is non self-embedding. Thus, the nonterminals $A_1, A_2, \ldots, A_p$ can be ordered such that every production with left-hand side A_i only contains nonterminals A_j with $i > j$. Thus, we show by induction that every derivation consists of at most $\sum_{i=0}^{p-1} k^i \leq k^p$ steps. Since $p \leq |G| \leq |L|$ and each derivation step can add at most ℓ new terminals, any word derivable in G has length at most $\ell \cdot k^{|L|}$. □

Theorem 5. *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. The case $k = 1$ follows from Lemmas 3 and 7, since every X-grammar covering L is an X-grammar generating a finite language $L' \supseteq L$ that satisfies $\mathsf{Xc}(L') \leq |L'| \leq 2^{|L|-1}$ and $\max\{\,|w| \mid w \in L'\,\} \leq \ell \cdot |L|$. Similarly, the case $k \geq 2$ follows from Lemmas 5 and 8. Hence, the sets

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

and, for $k \geq 2$,

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

are finite. □

So, for a class of CFGs as in Theorem 5, determining the cover complexity of L boils down to computing the exact complexity on the finite set $\mathcal{S}_L$.

5 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*. Since the lower bounds are hard to show in the cover formulation, we have not yet been able to obtain lower bounds on union and concatenation for fixed alphabets. The only exceptions are union in the cases of strict regular and strict linear grammars as well as concatenation in the case of strict regular grammars. The results of this section are summarised in Figure 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 $\Delta = \Gamma \setminus \{CF\}$.

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

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

5.1 Intersection

Theorem 6. *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 $|G_i| = \min\{|G_1|, |G_2|\}$. □

Theorem 7. *Let $X \in \Delta$. Then there exists a finite alphabet Σ such that for all $m, n \geq 1$, there are $L_1, L_2 \in \mathcal{P}_{\text{fin}}(\Sigma^*)$ with $\mathsf{Xcc}(L_1) \geq m$ and $\mathsf{Xcc}(L_2) \geq n$ 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, $m, n \geq 1$. From Corollary 1, it follows that there are $L_1, L_2 \in \mathcal{P}_{\text{fin}}(\Sigma^*)$ with $\mathsf{Xcc}(L_1) \geq m$ and $\mathsf{Xcc}(L_2) \geq n$. Assume, w.l.o.g., $\mathsf{Xcc}(L_1) \leq \mathsf{Xcc}(L_2)$. Define $L_2' = L_1 \cup L_2$. Then $\mathsf{Xcc}(L_2') \geq \mathsf{Xcc}(L_2) \geq \mathsf{Xcc}(L_1)$, for otherwise there would be a grammar generating $L_2' \supseteq L_2$ with less than $\mathsf{Xcc}(L_2)$ productions. Thus, we clearly have $\mathsf{Xcc}(L_1 \cap L_2') = \mathsf{Xcc}(L_1) = \min\{\, \mathsf{Xcc}(L_1), \mathsf{Xcc}(L_2') \,\}$. □

5.2 Union

Theorem 8. *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 $|G_i| = \mathsf{Xcc}(L_i)$ s.t. $N_1 \cap N_2 = \emptyset$. Since G_i is minimal and non self-embedding, S_i does not occur on the right-hand side of a production in P_i. Let $S \notin N_1 \cup N_2$ and $G = (N_1 \cup N_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P, S)$ where

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

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

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

Proof. Let $m, n \geq 1$. Then define $\Sigma = \{a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_n\}$, $L_1 = \{a_1, a_2, \ldots, a_m\}$, and $L_2 = \{b_1, b_2, \ldots, b_n\}$. Consequently, $L_1 \cup L_2 = \Sigma$ and, clearly, $\mathsf{Xcc}(L_1) = m$, $\mathsf{Xcc}(L_2) = n$, and the language $L_1 \cup L_2$ can only be covered by a trivial grammar. Therefore, $\mathsf{Xcc}(L_1 \cup L_2) = m + n = \mathsf{Xcc}(L_1) + \mathsf{Xcc}(L_2)$. □

Now, we prove—with respect to a fixed alphabet—a lower bound on the strict regular and 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 9. *Let $L \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*)$ and $\ell = \max\{\, |w| \mid w \in L \,\}$. Then*

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

Proof Sketch. First, show by induction on the length k of a derivation of $v \in \Sigma^*$ that $k \geq \left\lfloor \frac{|v|}{2} + 1 \right\rfloor$. Since in a strict linear grammar all right-hand sides of productions contain at most one nonterminal, no production can occur twice in such a derivation, for otherwise the generated language would be infinite. As a consequence, such a derivation uses k distinct productions in order to derive v. Thus, for some $w \in \Sigma^*$ with $|w| = \ell$, we have $k \geq \left\lfloor \frac{\ell}{2} + 1 \right\rfloor$. The SREG-case can be shown using similar arguments. □

Theorem 10. *Let $X \in \{SREG, SLIN\}$. Then there exists a finite alphabet Σ such that for all $m, n \geq 1$, there are $L_1, L_2 \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*)$ with $\mathsf{Xcc}(L_1) = m$ and $\mathsf{Xcc}(L_2) = n$ such that*

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

Proof. For $X = SREG$, let $\Sigma = \{a, b\}$ and, for $m, n \geq 1$, we define the finite languages $L_1 = \{a^m\}$ and $L_2 = \{b^n\}$. Moreover, let $L = L_1 \cup L_2$. Then, from Lemma 9, we get that $\mathsf{SREGcc}(L_1) \geq m$ and $\mathsf{SREGcc}(L_2) \geq n$. It is easy to see that also $\mathsf{SREGcc}(L_1) \leq m$ and $\mathsf{SREGcc}(L_2) \leq n$. 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 $\mathsf{SREGcc}(L) = \mathsf{SREGcc}(L_1 \cup L_2) \geq \mathsf{SREGcc}(L_1) + \mathsf{SREGcc}(L_2)$.

For $X = SLIN$, let $L_1 = \{a^{2m-1}\}$ and $L_2 = \{b^{2n-1}\}$ and define $L = L_1 \cup L_2$. Then proceed analogous to the SREG-case using Lemma 9. □

5.3 Concatenation

Theorem 11. *Let $X \in \{SREG, REG\}$ and $L_1, L_2 \in \mathcal{P}_{\text{fin}}(\Sigma^*)$. 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 Sketch. 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, without loss of generality, that $N_1 \cap N_2 = \emptyset$. First, note that in a right-linear and left-linear grammar all productions of the form $A \to w$ with $w \in \Sigma^*$ are used to derive the postfixes and prefixes of words, respectively.

For $X \in \{SREG, REG\}$, we construct an X-grammar covering L_1L_2 by taking the union $P_1 \cup P_2$ and replacing all productions of the form $A \to w \in P_1$ with $w \in \Sigma^*$ by $A \to wS_2$. Consequently, $\mathsf{Xcc}(L_1L_2) \leq \mathsf{Xcc}(L_1) + \mathsf{Xcc}(L_2)$.

For $X \in \{SLIN, LIN\}$, let $G_{(\mathsf{S})REG,i}$ and $G_{(\mathsf{S})LIN,i}$ be minimal $(\mathsf{S})REG$- and $(\mathsf{S})LIN$-grammars covering L_i, for $i \in \{1, 2\}$. Assume that these grammars have pairwise disjoint sets of nonterminals. We define two $(\mathsf{S})LIN$-grammars G_1 and G_2 covering L_1L_2 as follows: G_1 is obtained by taking the union $P_{(\mathsf{S})REG,1} \cup P_{(\mathsf{S})LIN,2}$ and replacing all productions of the form $A \to w \in P_{(\mathsf{S})REG,1}$ with $w \in \Sigma^*$ by $A \to wS_{(\mathsf{S})LIN,2}$. Similarly, G_2 is obtained by taking the union $P_{(\mathsf{S})LIN,1} \cup P_{(\mathsf{S})REG,2}$ and replacing all productions of the form $A \to w \in P_{(\mathsf{S})REG,2}$ with $w \in \Sigma^*$ by $A \to S_{(\mathsf{S})LIN,1}w$. Then simply take the grammar with the fewest number of productions out of G_1 and G_2. Thus, $\mathsf{Xcc}(L_1L_2) \leq \min\{\, (\mathsf{S})\mathsf{REGcc}(L_1) + (\mathsf{S})\mathsf{LINcc}(L_2), (\mathsf{S})\mathsf{LINcc}(L_1) + (\mathsf{S})\mathsf{REGcc}(L_2) \,\}$. □

The following lemma shows that a grammar covering the concatenation of two disjoint alphabets (where each contains 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 12.

Lemma 10. *Let $\Sigma = \Sigma_1 \uplus \Sigma_2$ with $|\Sigma_1|, |\Sigma_2| \geq 2$. Then for all* CFGs *$G$ with $L(G) \supseteq \Sigma_1\Sigma_2$, we have $|G| \geq |\Sigma_1| + |\Sigma_2|$.*

Proof Sketch. Proceed by induction on $|\Sigma|$, making a case distinction in the base case $|\Sigma| = 4$ and reducing the step case to the induction hypothesis by deleting productions that contain the new letter. □

Theorem 12. *Let $X \in \{SREG, REG\}$. Then, for all $m, n \geq 2$, there is a finite alphabet Σ and $L_1, L_2 \in \mathcal{P}_{\text{fin}}(\Sigma^*)$ with $\mathsf{Xcc}(L_1) = m$ and $\mathsf{Xcc}(L_2) = n$ s.t.*

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 $m, n \geq 2$ and define the alphabet $\Sigma = \{a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_n\}$ as well as the languages $L_1 = \{a_1, a_2, \ldots, a_m\}$, $L_2 = \{b_1, b_2, \ldots, b_n\}$, and let $X \in \{SREG, REG\}$. Then clearly we have $\mathsf{Xcc}(L_1) = m$ and $\mathsf{Xcc}(L_2) = n$. Thus,

since every X-grammar is context-free, we have by Lemma 10 that $\mathsf{Xcc}(\Sigma) = \mathsf{Xcc}(L_1L_2) \geq m + n = \mathsf{Xcc}(L_1) + \mathsf{Xcc}(L_2)$ and $(\mathsf{S})\mathsf{LINcc}(L_1L_2) \geq m + n = \min\{(\mathsf{S})\mathsf{REGcc}(L_1) + (\mathsf{S})\mathsf{LINcc}(L_2), (\mathsf{S})\mathsf{LINcc}(L_1) + (\mathsf{S})\mathsf{REGcc}(L_2)\}$. □

Theorem 13. *There exists a finite alphabet Σ such that for all $m, n \geq 1$, there exist $L_1, L_2 \in \mathcal{P}_{\mathrm{fin}}(\Sigma^*)$ with $\mathsf{SREGcc}(L_1) = m$ and $\mathsf{SREGcc}(L_2) = n$ such that*

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

Proof. Let $\Sigma = \{a\}$ and, for $m, n \geq 1$, define $L_1 = \{a^m\}$ and $L_2 = \{a^n\}$. From Lemma 9, we get $\mathsf{SREGcc}(L_1) \geq m$ and $\mathsf{SREGcc}(L_2) \geq n$. It is easy to see that also $\mathsf{SREGcc}(L_1) \leq m$ and $\mathsf{SREGcc}(L_2) \leq n$. Again, by Lemma 9, it follows that $\mathsf{SREGcc}(L_1L_2) \geq m + n = \mathsf{SREGcc}(L_1) + \mathsf{SREGcc}(L_2)$. □

6 Conclusion

In this paper, we have investigated cover complexity measures for finite languages on three different levels and shown that every complexity measure on finite languages naturally induces a corresponding cover complexity measure. We have characterised in which situations 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 (see Figure 1). Generalising the incompressibility result of [10, 11] in a suitable fashion seems to be a promising starting point for improving the lower bounds from growing to fixed alphabets. 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.

Acknowledgements. The authors would like to thank Markus Holzer and the anonymous reviewers for several useful comments and suggestions concerning the results in this paper.

References

1. Alspach, B., Eades, P., Rose, G.: A Lower-bound for the Number of Productions Required for a Certain Class of Languages. Discrete Applied Mathematics **6**(2), 109–115 (1983)
2. Bucher, W.: A Note on a Problem in the Theory of Grammatical Complexity. Theoretical Computer Science **14**, 337–344 (1981)
3. Bucher, W., Maurer, H.A., II, K.C.: Context-free Complexity of Finite Languages. Theoretical Computer Science **28**, 277–285 (1984)

4. Bucher, W., Maurer, H.A., II, K.C., Wotschke, D.: Concise Description of Finite Languages. Theoretical Computer Science **14**, 227–246 (1981)
5. Câmpeanu, C., Santean, N., Yu, S.: Minimal cover-automata for finite languages. In: Champarnaud, J., Maurel, D., Ziadi, D. (eds.) International Workshop on Implementing Automata (WIA'98). Lecture Notes in Computer Science, vol. 1660, pp. 43–56. Springer (1998)
6. Câmpeanu, C., Santean, N., Yu, S.: Minimal cover-automata for finite languages. Theoretical Computer Science **267**(1-2), 3–16 (2001)
7. Cerný, A.: Complexity and minimality of context-Free grammars and languages. In: Gruska, J. (ed.) Mathematical Foundations of Computer Science (MFCS) 1977. Lecture Notes in Computer Science, vol. 53, pp. 263–271. Springer (1977)
8. Dassow, J.: Descriptional Complexity and Operations—Two Non-classical Cases. In: Pighizzini, G., Câmpeanu, C. (eds.) Workshop on Descriptional Complexity of Formal Systems (DCFS). Lecture Notes in Computer Science, vol. 10316, pp. 33–44. Springer, Milano, Italy (2017)
9. Dassow, J., Harbich, R.: Production Complexity of Some Operations on Context-Free Languages. In: Kutrib, M., Moreira, N., Reis, R. (eds.) Workshop on Descriptional Complexity of Formal Systems (DCFS). Lecture Notes in Computer Science, vol. 7386, pp. 141–154. Springer, Braga, Portugal (2012)
10. Eberhard, S., Hetzl, S.: Compressibility of Finite Languages by Grammars. In: Shallit, J., Okhotin, A. (eds.) Descriptional Complexity of Formal Systems (DCFS) 2015. Lecture Notes in Computer Science, vol. 9118, pp. 93–104. Springer (2015)
11. Eberhard, S., Hetzl, S.: On the compressibility of finite languages and formal proofs. Information and Computation **259**, 191–213 (2018)
12. Gruska, J.: On a Classification of Context-Free Languages. Kybernetika **3**(1), (22)–29 (1967)
13. Gruska, J.: Some Classifications of Context-Free Languages. Information and Control **14**(2), 152–179 (1969)
14. Gruska, J.: Complexity and unambiguity of context-free grammars and languages. Information and Control **18**(5), 502–519 (1971)
15. Gruska, J.: On the Size of Context-free Grammars. Kybernetika **8**(3), 213–218 (1972)
16. Hetzl, S.: Applying Tree Languages in Proof Theory. In: Dediu, A.H., Martín-Vide, C. (eds.) Language and Automata Theory and Applications (LATA) 2012. Lecture Notes in Computer Science, vol. 7183, pp. 301–312. Springer (2012)
17. Hetzl, S., Leitsch, A., Reis, G., Tapolczai, J., Weller, D.: Introducing Quantified Cuts in Logic with Equality. In: Demri, S., Kapur, D., Weidenbach, C. (eds.) Automated Reasoning - 7th International Joint Conference, IJCAR. Lecture Notes in Computer Science, vol. 8562, pp. 240–254. Springer (2014)
18. Hetzl, S., Leitsch, A., Reis, G., Weller, D.: Algorithmic introduction of quantified cuts. Theoretical Computer Science **549**, 1–16 (2014)
19. Hetzl, S., Leitsch, A., Weller, D.: Towards Algorithmic Cut-Introduction. In: Logic for Programming, Artificial Intelligence and Reasoning (LPAR-18). Lecture Notes in Computer Science, vol. 7180, pp. 228–242. Springer (2012)
20. Pudlák, P.: Twelve Problems in Proof Complexity. In: Hirsch, E.A., Razborov, A.A., Semenov, A.L., Slissenko, A. (eds.) Third International Computer Science Symposium in Russia (CSR). Lecture Notes in Computer Science, vol. 5010, pp. 13–27. Springer (2008)
21. Tuza, Z.: On the Context-Free Production Complexity of Finite Languages. Discrete Applied Mathematics **18**(3), 293–304 (1987)