Boolean Kernels of Context-Free Languages

Martin Kutrib[1](✉) and Luca Prigioniero[2]

[1] Institut für Informatik, Universität Giessen,
Arndtstr. 2, 35392 Giessen, Germany
kutrib@informatik.uni-giessen.de

[2] Dipartimento di Informatica, Università degli Studi di Milano,
Via Celoria 18, 20133 Milan, Italy
prigioniero@di.unimi.it

Abstract. While the closure of a language family $\mathscr{L}$ under certain language operations is the least family of languages which contains all members of $\mathscr{L}$ and is closed under all of the operations, a kernel of $\mathscr{L}$ is a maximal family of languages which is a sub-family of $\mathscr{L}$ and is closed under all of the operations. Here we investigate properties of the Boolean kernels of the family of context-free languages. Additionally, languages that are mandatory for each Boolean kernel and languages that are optional for Boolean kernels are studied. That is, we consider the intersection of all Boolean kernels as well as their union. The expressive capacities of these families are addressed leading to a hierarchical structure. Further closure properties are considered. Furthermore, we study descriptional complexity aspects of these families, where languages are represented by context-free grammars with proofs attached. It turns out that the size trade-offs between all families in question and deterministic context-free languages are non-recursive. That is, one can choose an arbitrarily large recursive function f, but the gain in economy of description eventually exceeds f when changing from the latter system to the former.

1 Introduction

Classical and well-developed concepts to represent (formal) languages are, for example, grammars, language equations, or accepting automata. Similarly, families of languages can be represented in several ways. For example, a language family can be defined to be the set of all languages represented by a certain type of grammar, automaton model, language equation, or by applying appropriate operations on other language families. From a practical point of view, there is often a considerable interest in language families that are robust with respect to language operations, that is, the families are preferably closed under the operations, and/or in language families that admit efficient recognizers. A good example are context-free languages, that are one of the most important and most developed area of formal language theory. However, the family is not closed under the two Boolean operations complementation and intersection. Moreover,

S. Maneth (Ed.): CIAA 2021, LNCS 12803, pp. 152–164, 2021.
https://doi.org/10.1007/978-3-030-79121-6_13

the known upper bound on the time complexity for context-free language recognition still exceeds $O(n^2)$. As an approach to characterize language families having strong closure properties and efficient recognizers but decrease the expressive capacity only slightly, closures of sub-classes of the context-free languages have been investigated. The Boolean closure of the linear context-free languages offers a significant increase in expressive capacity compared with the linear context-free languages itself. In addition, it preserves the attractively efficient recognition algorithm [10] taking $O(n^2)$ time and $O(n)$ space. The systematic investigation of the Boolean closures of arbitrary and deterministic context-free languages started in [12–14]. The closure of deterministic languages under the regular operations is studied in [1], while the regular closure of the linear context-free languages is considered in [9].

Here we are interested in language families with strong closure properties obtained as sub-families of a given family instead of closing and, thus, extending the family. To this end, we study Boolean kernels of the family of context-free languages. Basically, such a kernel is a maximal sub-family of the context-free languages that is closed under the Boolean operations.

The paper is organized as follows. After presenting the basic definitions and notions in the next section, Sect. 3 deals with the expressive capacities of Boolean kernels of context-free languages as well as with languages that are mandatory for each Boolean kernel and languages that are optional for Boolean kernels. For the latter, the intersection and union of all Boolean kernels is considered. The hierarchical structure of these families is depicted in Fig. 1. Section 4 is devoted to additional closure properties. In particular, the operations reversal, concatenation, and inverse homomorphism are studied. The results are summarized in Table 1. Descriptional complexity aspects are dealt with in Sect. 5. The size of a language is given by the size of its representation. Since, in most cases, no automata or grammar characterizations are known for kernels, here we use representations by context-free grammars which come with a corresponding proof attached. The proofs certify that the grammar generates a language belonging to the desired sub-family. The length of the proof is then added to the size of the grammar. It turns out that the size trade-offs between all families in question and deterministic context-free languages are non-recursive. That is, one can choose an arbitrarily large recursive function f, but the gain in economy of description eventually exceeds f when changing from the latter system to the former. Finally, we discuss some interesting untouched problems and questions for further research in Sect. 6.

2 Preliminaries

We write Σ^* for the set of all words over a finite alphabet Σ. The *empty word* is denoted by λ, and we set $\Sigma^+ = \Sigma^* \setminus \{\lambda\}$. The *reversal* of a word w is denoted by w^R, and for the *length* of w we write $|w|$. For the number of *occurrences* of a symbol a in w we use the notation $|w|_a$. Set *inclusion* is denoted by $\subseteq$ and *strict set inclusion* by $\subset$.

A subset of Σ^* is called a *(formal) language* over Σ. A *language operation* is an operation whose finite number of parameters are languages, and whose result is a language. For example, the *complement* of a language is defined with respect to the underlying alphabet Σ. That is, the complement of $L \subseteq \Sigma^*$ is $\overline{L} = \{\, w \in \Sigma^* \mid w \notin L \,\}$. For all $k \geq 1$, a k-ary language operation $\circ$ is said to be *idempotent* if $\circ(L, L, \ldots, L) = L$, for all L in the domain of $\circ$. For easier writing, here we call even a unary language operation $\circ$ with the property $\circ(L) = L$ idempotent (so we do *not* require $\circ(\circ(L)) = \circ(L)$).

Let Ω be an infinite enumerable set of letters. The set $\mathscr{L}$ is a *family of languages* over Ω if for each $L \in \mathscr{L}$ there is a finite subset $\Sigma \subset \Omega$ such that $L \subseteq \Sigma^*$. In the sequel we tacitly omit Ω when it is understood.

Let $\mathscr{L}$ be a family of languages and $op_1, op_2, \ldots, op_k$, $k \geq 1$, be a finite number of operations defined on $\mathscr{L}$.

1. By $\Gamma_{op_1,op_2,\ldots,op_k}(\mathscr{L})$ we denote the $(op_1, op_2, \ldots, op_k)$ *closure of* $\mathscr{L}$. That is, the *least family of languages which contains all members of* $\mathscr{L}$ *and is closed under* $op_1, op_2, \ldots, op_k$. In other words, there exists no language family $\mathscr{L}'$ that is closed under $op_1, op_2, \ldots, op_k$ such that $\mathscr{L} \subseteq \mathscr{L}' \subset \Gamma_{op_1,op_2,\ldots,op_k}(\mathscr{L})$.
2. By $\gamma_{op_1,op_2,\ldots,op_k}(\mathscr{L})$ we denote the set of $(op_1, op_2, \ldots, op_k)$ *kernels of* $\mathscr{L}$. That is, the set of *maximal families of languages which are sub-families of* $\mathscr{L}$ *and are closed under* $op_1, op_2, \ldots, op_k$. In other words, for all kernels $\kappa \in \gamma_{op_1,op_2,\ldots,op_k}(\mathscr{L})$ there exists no language family $\mathscr{L}'$ that is closed under $op_1, op_2, \ldots, op_k$ such that $\kappa \subset \mathscr{L}' \subseteq \mathscr{L}$.

In particular, we consider the operations complementation ($\overline{\ }$), union ($\cup$), and intersection ($\cap$), which are called *Boolean operations.* Accordingly, we write Γ_{BOOL} for $\Gamma_{-,\cup,\cap}$ and γ_{BOOL} for $\gamma_{-,\cup,\cap}$.

Since special attention is paid to sub-classes of context-free languages, we refer to the literature, for example to [3], for detailed definitions of context-free grammars and of the characterizing automata models. In particular, an automaton model for the recognition of context-free languages is the nondeterministic pushdown automaton. Its deterministic variant characterizes the deterministic context-free languages (DCFL).

It is known from [8] that the sets $\gamma_{\text{BOOL}}(\text{CFL})$ as well as $\gamma_{\text{BOOL}}(\text{DCFL})$ include infinitely many kernels, while the complementation kernel of the context-free languages is unique. Moreover, not all context-free languages belong to some Boolean kernel, while any deterministic context-free language belongs to some kernel $\kappa \in \gamma_{\text{BOOL}}(\text{DCFL})$.

3 Expressive Capability

In connection with the question of whether any language of a family belongs to some kernel based on given operations, or whether there are languages that do not belong to any of such kernels, the union of all kernels has been considered. Similarly, the question which languages belong to all kernels based on given operations raised the definition of the intersection of all of these kernels.

The union of all Boolean kernels of the context-free languages is denoted by $\mathscr{U}$, that is, $\mathscr{U} = \{ L \mid L \in \kappa \text{ for some } \kappa \in \gamma_{\mathrm{BOOL}}(\mathrm{CFL}) \}$.

Similarly, the intersection of all Boolean kernels of the context-free languages is denoted by $\mathscr{I}$, that is, $\mathscr{I} = \{ L \mid L \in \kappa \text{ for all } \kappa \in \gamma_{\mathrm{BOOL}}(\mathrm{CFL}) \}$.

It turns out that the union of all Boolean kernels of the context-free languages characterizes an interesting language family. Theorem 1 shows that it coincides with the unique complementation kernel in $\gamma\_(\mathrm{CFL})$. That is interesting in itself but beyond that, the unique complementation kernel is also known as the family of strongly context-free languages [7]. A machine characterization of that family in terms of self-verifying pushdown automata is obtained in [2].

Theorem 1. *The family $\mathscr{U}$ coincides with the unique complementation kernel in $\gamma\_(\mathrm{CFL})$.*

Proof. Let L be a language in $\mathscr{U} \subseteq \mathrm{CFL}$. Then there is a $\kappa \in \gamma_{\mathrm{BOOL}}(\mathrm{CFL})$ with $L \in \kappa$. Since κ is closed under complementation, the complement $\overline{L}$ of L belongs to κ as well. We conclude that $\overline{L}$ belongs to $\mathscr{U}$ and, thus, to CFL. In particular, since L and $\overline{L}$ are context free, they belong to the unique kernel in $\gamma\_(\mathrm{CFL})$.

For the converse, let L be some language over the alphabet Σ such that L and, thus, $\overline{L}$ belong to the unique kernel in $\gamma\_(\mathrm{CFL})$. We consider the set $\nu = \{L, \overline{L}, \Sigma^*, \emptyset\}$ which is clearly closed under complementation, union, and intersection. Since L and $\overline{L}$ are context free, either ν is itself a Boolean kernel of CFL, or there exists a kernel in $\gamma_{\mathrm{BOOL}}(\mathrm{CFL})$ having ν, and thus $\{L\}$, as subset. So, L belongs to $\mathscr{U}$. □

Since the family of context-free languages is not closed under complementation but by Theorem 1 the family $\mathscr{U}$ is, the inclusion $\mathscr{U} \subset \mathrm{CFL}$ is strict. Moreover, since there are infinitely many different Boolean kernels in $\gamma_{\mathrm{BOOL}}(\mathrm{CFL})$, the maximality of kernels implies that any $\kappa \in \gamma_{\mathrm{BOOL}}(\mathrm{CFL})$ is strictly included in $\mathscr{U}$. For example, consider the two context-free languages $L_1 = \{ a^n b^n a^m \mid m, n \geq 1 \}$ and $L_2 = \{ a^m b^n a^n \mid m, n \geq 1 \}$. Their complements are context free as well and, thus, both belong to the unique kernel in $\gamma\_(\mathrm{CFL})$ which coincides with $\mathscr{U}$. Therefore, by Theorem 1 both belong to some Boolean kernel from $\gamma_{\mathrm{BOOL}}(\mathrm{CFL})$. However, languages L_1 and L_2 cannot belong to the same Boolean kernel from $\gamma_{\mathrm{BOOL}}(\mathrm{CFL})$, since their intersection is the non-context-free language $\{ a^n b^n a^n \mid n \geq 1 \}$.

In order to continue with the exploration of the hierarchical structure of Boolean kernels, we turn to consider the family $\mathscr{I}$ which is the intersection of all Boolean kernels of the context-free languages.

Proposition 2. *The family $\mathscr{I}$ is strictly included in any Boolean kernel $\kappa \in \gamma_{\mathrm{BOOL}}(\mathrm{CFL})$.*

It is shown in [8] that all Boolean kernels of the context-free languages include the regular languages and some non-regular languages. So far, we have the hierarchy $\mathrm{REG} \subset \mathscr{I} \subset \kappa \subset \mathscr{U} \subset \mathrm{CFL}$, for all kernels $\kappa \in \gamma_{\mathrm{BOOL}}(\mathrm{CFL})$, (see Fig. 1). Finally, we turn to compare the family of deterministic context-free languages with the hierarchical structure of Boolean kernels.

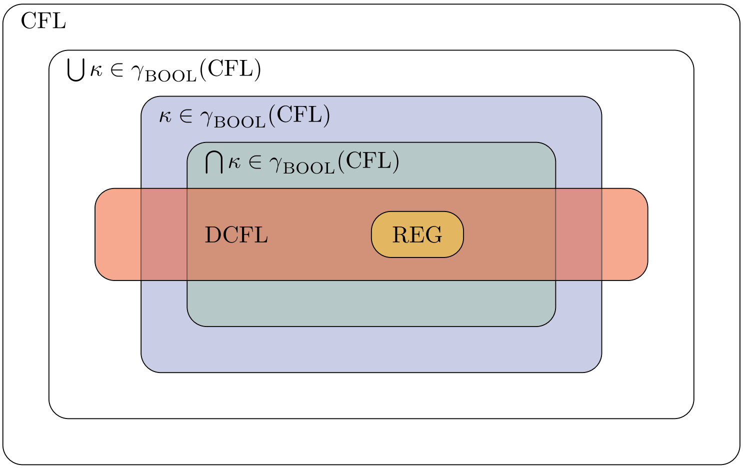
CFL
$\bigcup \kappa \in \gamma_{\text{BOOL}}(\text{CFL})$
$\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$
$\bigcap \kappa \in \gamma_{\text{BOOL}}(\text{CFL})$
DCFL
REG

Fig. 1. Hierarchical structure of language classes. The class $\bigcup \kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ denotes the union of all Boolean kernels of CFL. It coincides with the unique complementation kernel of CFL. By $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ we denote an arbitrary Boolean kernel of CFL, and by $\bigcap \kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ the intersection of all Boolean kernels of CFL.

First we deduce that the family DCFL is strictly included in the family $\mathscr{U}$.

Corollary 3. *The family* DCFL *is strictly included in the family* $\mathscr{U}$*.*

Proof. By Theorem 1, any context-free language whose complement is also context free belongs to $\mathscr{U}$. Since the family DCFL is closed under complementation and a subset of CFL, we obtain the inclusion DCFL $\subseteq \mathscr{U}$. Its strictness is witnessed, for example, by the context-free language $\{\, w \in \{a,b\}^* \mid w = w^R \,\}$ not belonging to DCFL whose complement is also context free (cf. [2]). So, it belongs to $\mathscr{U}$ but is not deterministic context free. □

Concerning an arbitrary Boolean kernel $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ and the family $\mathscr{I}$ we obtain incomparability with DCFL.

Theorem 4. *For any* $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$*, the family* DCFL *is incomparable with* κ *and with the family* $\mathscr{I}$*.*

Proof. Both languages $\{\, a^n b^n a^m \mid m,n \geq 1 \,\}$ and $\{\, a^m b^n a^n \mid m,n \geq 1 \,\}$ are deterministic context free. Assume that they do belong to κ. Since κ is closed under intersection, the non-context-free language $\{\, a^n b^n a^n \mid n \geq 1 \,\}$ must belong to κ as well, a contradiction. So, there is a language in DCFL $\setminus \kappa$ and, trivially, in DCFL $\setminus \mathscr{I}$.

Conversely, it is known that the languages $L_1 = \{\, a^n b^n \mid n \geq 1 \,\}$ and $L_2 = \{\, a^n b^{2n} \mid n \geq 1 \,\}$ are included in any Boolean kernel of the context-free languages [8]. So, they belong to the family $\mathscr{I}$ and, trivially, to κ. We consider the union $L_1 \cup L_2$. Since L_1 and L_2 belong to any Boolean kernel of the context-free languages which, in turn are closed under union, also $L_1 \cup L_2$ must belong to any Boolean kernel of the context-free languages. In other words, $L_1 \cup L_2 = \{\, a^n b^m \mid m = n \text{ or } m = 2n, m, n \geq 1 \,\}$ belongs to $\mathscr{I}$ and, trivially, to κ. But $L_1 \cup L_2$ is not deterministic context free. So, there is a language in $\kappa \setminus \text{DCFL}$ and, trivially, in $\mathscr{I} \setminus \text{DCFL}$. □

So far we have derived the comparisons of DCFL with the other families in question. However, as shown in Fig. 1, its position in the hierarchical structure needs a finer adjustment. The first question is whether the union of some kernel κ and DCFL already characterizes the family $\mathscr{U}$. Proposition 5 gives a negative answer.

Proposition 5. *For any $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$, there is a language in $\mathscr{U} \setminus (\kappa \cup \text{DCFL})$.*

To continue with the finer adjustment let us next ask whether the union of DCFL and $\mathscr{I}$ already captures the kernels κ. Again, the answer is negative.

Proposition 6. *There exists a $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ such that there is a language in $\kappa \setminus (\mathscr{I} \cup \text{DCFL})$.*

Proof. A language we are looking for has been considered in the proof of Proposition 5. There, it is shown that the complement of the context-free language $L_1 = \{\, a^n b^n a^m \mid m, n \geq 1 \,\} \cup \{\, a^n b^{2n} a^m \mid m, n \geq 1 \,\}$ is context free as well, and, thus, that L_1 belongs to $\mathscr{U}$. This implies that L_1 belongs to some kernel $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$.

Similarly, it is shown that $L_2 = \{\, a^m b^n a^n \mid m, n \geq 1 \,\} \cup \{\, a^m b^n a^{2n} \mid m, n \geq 1 \,\}$ belongs to some kernel from $\gamma_{\text{BOOL}}(\text{CFL})$, but L_1 and L_2 cannot belong to the same kernel.

So, we conclude that at least one of L_1 and L_2 does not belong to $\mathscr{I}$. Both languages are not deterministic context free. So, the assertion follows. □

The last two areas to be considered in Fig. 1 are the intersection of DCFL and κ without $\mathscr{I}$, and the intersection of DCFL and $\mathscr{I}$ without REG. For the latter, we can utilize once more the non-regular but deterministic context-free language $\{\, a^n b^n \mid n \geq 1 \,\}$ that is included in any Boolean kernel of the context-free languages [8].

Corollary 7. *There is a language in $(\text{DCFL} \cap \mathscr{I}) \setminus \text{REG}$.*

For the former, we have the following result.

Proposition 8. *There exists a $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ such that there is a language in $(\text{DCFL} \cap \kappa) \setminus \mathscr{I}$.*

4 (Non-)Closure Properties

The closure properties of the kernels from $\gamma_{\text{BOOL}}(\text{CFL})$ under the Boolean operations are trivial by definition. By Theorem 1, the properties of $\mathscr{U}$ can be derived from the results on strongly context-free languages obtained in [7]. In particular, it is closed under complementation but is not closed under union and intersection (see Table 1). For the family $\mathscr{I}$ and Boolean operations we have the following situation.

Proposition 9. *The family $\mathscr{I}$ is closed under complementation, union, and intersection.*

Since all $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ include the regular languages and are closed under intersection, they are closed under intersection with regular sets. The same argument applies to the family $\mathscr{I}$.

We call a language that witnesses the non-inclusion of another language in some family by violating closure properties *toxic*. More precisely, let $\mathscr{L}$ be some family of languages not closed under an operation $\circ$, and L be a language belonging to $\mathscr{L}$. Then a language $L' \in \mathscr{L}$ is said to be $\mathscr{L}$-$\circ$-*toxic for* L if and only if $L \circ L' \notin \mathscr{L}$.

Lemma 10. *Let $\mathscr{L}$ be a family of languages that includes $\emptyset$ and Σ^*, for all alphabets Σ, and $\kappa \in \gamma_{\text{BOOL}}(\mathscr{L})$. A language $L \in \mathscr{L}$ does not belong to κ if and only if either $\overline{L} \notin \mathscr{L}$ or there is a language $L' \in \kappa$ that is $\mathscr{L}$-$\cap$-toxic or $\mathscr{L}$-$\cup$-toxic for L.*

Proof. Let $L \in \mathscr{L}$ be a language not belonging to κ. Assume that $\overline{L} \in \mathscr{L}$ and all languages $L' \in \kappa$ are neither $\mathscr{L}$-$\cap$-toxic nor $\mathscr{L}$-$\cup$-toxic for L. Then we consider $\Gamma_{\text{BOOL}}(\kappa \cup \{L, \overline{L}\})$. In particular, we have that L as well as $\overline{L}$ belong to $\mathscr{L}$, $L \cap L'$ and $L \cup L'$ do belong to $\mathscr{L}$ for *all* $L' \in \kappa$. Moreover, $L \cap \overline{L} = \emptyset \in \kappa$ and $L \cup \overline{L} = \Sigma^* \in \kappa$. Therefore, $\Gamma_{\text{BOOL}}(\kappa \cup \{L, \overline{L}\})$ is included in $\mathscr{L}$. This contradicts the maximality of κ.

If $L \in \kappa$ then $\overline{L} \in \kappa \subseteq \mathscr{L}$, since κ is closed under complementation. Moreover, since κ is closed under union and intersection, for all languages $L' \in \kappa$, we have $L \cup L' \in \kappa \subseteq \mathscr{L}$ and $L \cap L' \in \kappa \subseteq \mathscr{L}$. We conclude that L' is neither $\mathscr{L}$-$\cup$-toxic nor $\mathscr{L}$-$\cap$-toxic for L. □

Reversal. The family $\mathscr{U}$ is closed under reversal. Before we turn to the closure of the family $\mathscr{I}$, we show that the closure may get lost for fixed kernels $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$.

Proposition 11. *There is a kernel $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ that is not closed under reversal.*

Theorem 12. *The family $\mathscr{I}$ is closed under reversal.*

Proof. Assume in contrast to the assertion that there is an $L \in \mathscr{I}$ such that $L^R \notin \mathscr{I}$. Then there is a $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ with $L^R \notin \kappa$.

Consider the complement $\overline{L^R}$. Since complementation commutes with reversal we have $\overline{L^R} = \overline{L}^R$. Since $L \in \kappa$ we derive $\overline{L} \in \kappa$ and, since CFL is closed under reversal also $\overline{L}^R = \overline{L^R} \in \text{CFL}$. Knowing this we apply Lemma 10 and conclude that there is a language $L' \in \kappa$ that is CFL-$\cap$-toxic or CFL-$\cup$-toxic for L^R. Since CFL is closed under union, L' must be CFL-$\cap$-toxic for L^R.

So, $L^R \cap L'$ does not belong to CFL. Since the family CFL is closed under reversal, we conclude $(L^R \cap L')^R \notin \text{CFL}$ and, thus, $(L^R)^R \cap (L')^R = L \cap (L')^R \notin \text{CFL}$. Since L belongs to *all* Boolean kernels of CFL, we derive that all Boolean kernels do not include $(L')^R$. Now Theorem 1 implies that either $(L')^R$ or $\overline{(L')^R}$ is not context free. Again, since complementation commutes with reversal we obtain that either $(L')^R$ or $\overline{L'}^R$ is not context free. However, $L' \in \kappa$ implies $\overline{L'} \in \kappa$. By the closure of CFL under reversal we obtain the contradiction $(L')^R \in \text{CFL}$ and $\overline{L'}^R \in \text{CFL}$. □

Concatenation and Inverse Homomorphism. In order to prove the non-closure of the family $\mathscr{I}$ and *all* kernels $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ under concatenation and inverse homomorphism, we consider semilinear languages that are subsets of a^*b^*, where the number of b's depends linearly on the number of a's. The dependency is given by linear functions $\varphi\colon \mathbb{N} \to \mathbb{N}$ with $\varphi(n) = c_1 \cdot n + c_0$, for some $c_0, c_1 \geq 0$. For such functions, we define $L_\varphi = \{\, a^n b^{\varphi(n)} \mid n \geq 0 \,\}$. In [8] it has been shown that all regular languages as well as all languages L_φ belong to all Boolean kernels of CFL. A generalization of the proofs reveals that this is true also for all reversals of the languages L_φ, that is, all languages L_φ^R belong to all Boolean kernels of CFL.

Theorem 13. *The family $\mathscr{I}$ and all kernels $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ are not closed under concatenation, not even with concatenation of unary regular sets.*

Theorem 14. *The family $\mathscr{I}$ and all kernels $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ are not closed under inverse homomorphisms, not even under length-preserving inverse homomorphisms.*

Proof. Let $\varphi\colon \mathbb{N} \to \mathbb{N}$ be the linear function $\varphi(n) = 2n$. We know that the languages $L_\varphi = \{\, a^n b^{2n} \mid n \geq 0 \,\}$ and L_φ^R belong to all kernels $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$. Furthermore, let $h\colon \{a,b,c\}^* \to \{a,b\}^*$ be the homomorphism $h(a) = a$, $h(b) = h(c) = b$ and $\hat{h}\colon \{a,b,c\}^* \to \{a,b\}^*$ be the homomorphism $\hat{h}(a) = \hat{h}(b) = b$, $\hat{h}(c) = a$. Then we have

$$h^{-1}(L_\varphi) \cap a^*b^*c^* = \{\, a^n b^m c^k \mid n \geq 0, m+k = 2n \,\} \text{ and}$$
$$\hat{h}^{-1}(L_\varphi^R) \cap a^*b^*c^* = \{\, a^k b^m c^n \mid n \geq 0, m+k = 2n \,\}.$$

Assume that the family $\mathscr{I}$ or some kernel $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ is closed under inverse homomorphism. Since they include the regular languages and are closed under intersection, we derive that they include the language

$$h^{-1}(L_\varphi) \cap \hat{h}^{-1}(L_\varphi^R) \cap a^*b^*c^* = \{\, a^n b^n c^n \mid n \geq 0 \,\}$$

as well, a contradiction. □

Table 1. Closure properties of the language families discussed. Symbol • denotes concatenation and κ stands for an arbitrary but fixed kernel from $\gamma_{\text{BOOL}}(\text{CFL})$. The properties shown for κ hold for *all* $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ with the exception of reversal. For reversal, it has been shown that *some* kernels are not closed. It is currently open if there exists some $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ that is closed under reversal. The non-closure under inverse homomorphism holds even for length-preserving homomorphisms.

	$\overline{}$	$\cup$	$\cap R$	$\cap$	REV	$\bullet$	h^{-1}
CFL	✗	✓	✓	✗	✓	✓	✓
$\mathscr{U}$	✓	✗	✓	✗	✓	✗	✓
κ	✓	✓	✓	✓	✗	✗	✗
$\mathscr{I}$	✓	✓	✓	✓	✓	✗	✗
DCFL	✓	✗	✓	✗	✗	✗	✓
REG	✓	✓	✓	✓	✓	✓	✓

5 Descriptional Complexity

One topic in the field of descriptional complexity is to study the relative succinctness of different representations of languages by automata, grammars, and descriptional systems from a more abstract perspective. For languages that have more than one representation, the size trade-offs when changing the representation may be bounded by a recursive function or not. In the latter case we are faced with the phenomenon of so-called non-recursive trade-offs. In particular, whenever the trade-off from one descriptional system to another is non-recursive, one can choose an arbitrarily large recursive function f but the gain in economy of description eventually exceeds f when changing from the latter system to the former. See [6] for more details on descriptional complexity.

In order to deal with such questions for kernels, a descriptional system for languages from the kernel is necessary whose size can be measured. Since, in general, no automata or grammar characterizations are known for kernels, we take up an idea of Hartmanis [4] who raised the question whether the trade-off between two descriptional systems is caused by the fact that in one system it can be proved what is accepted, but that no such proofs are possible in the other system. For example, consider descriptional systems for the deterministic context-free languages. It is easy to verify whether a given pushdown automaton is deterministic, but there is no uniform way to verify that a nondeterministic pushdown automaton accepts a deterministic context-free language. So, one may ask whether the trade-off is affected if descriptional systems are considered which come with a corresponding proof attached whose length is added to the size of the system.

So, in the following we consider the representation of context-free languages by nondeterministic pushdown automata (NPDA) to which a proof is attached that the accepted language belongs to $\mathscr{U}$, $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$, or $\mathscr{I}$. We denote these automata as $\mathscr{U}$-NPDA, κ-NPDA, or $\mathscr{I}$-NPDA. The size of such an automaton is the length of the description of the automaton plus the length of the proof, say in binary. Then it is clear that, for any $c \geq 1$, there are only finitely many $\mathscr{U}$-NPDA, κ-NPDA, or $\mathscr{I}$-NPDA whose size is at most c.

It will turn out that the trade-offs between any of these three descriptional systems and deterministic pushdown automata (DPDA) are non-recursive. The proof is by reduction of the halting problem for Turing machines on empty tape. To this end, histories of Turing machine computations are encoded into strings. It suffices to consider deterministic Turing machines with one single tape and one single read-write head. Without loss of generality and for technical reasons, we safely may assume that the Turing machines cannot print blanks, can halt only after an odd number of moves, and accepts by halting. The size of a Turing machine is again measured as the length of its description. As for the NPDA, there are only finitely many Turing machines of the same size.

Let Q be the state set of some Turing machine M, where q_0 is the initial state, $T \cap Q = \emptyset$ is the tape alphabet containing the blank symbol, and $\Sigma \subset T$ is the input alphabet. Then a configuration of M can be written as a word of the form T^*QT^* such that $t_1t_2 \cdots t_iqt_{i+1} \cdots t_n$ is used to express that M is in state q, scanning tape symbol t_{i+1}, and t_1, t_2 to t_n is the support of the tape inscription.

Dependent on M we define the language of valid computations. Let $\$ \notin T \cup Q$, $n \geq 0$, and $w_i \in T^*QT^*$, $0 \leq i \leq 2n+1$, be configurations of M. Then $\text{VALC}(M)$ is defined to be the language of all words of the form

$$\$w_0\$w_1^R\$w_2\$w_3^R\$ \cdots \$w_{2n}\$w_{2n+1}^R\$,$$

where w_0 is an initial configuration of the form q_0, w_{2n+1} is a halting configuration, and w_i is the successor configuration of w_{i-1}, $1 \leq i \leq 2n+1$. The language of *invalid computations* $\text{INVALC}(M)$, is the complement of $\text{VALC}(M)$ with respect to the alphabet $\{\$\} \cup T \cup Q$.

Corollary 15. *For any deterministic Turing machine M, the language INVALC(M) is a linear context-free language, such that its nondeterministic one-turn pushdown automaton can effectively be constructed from M.*

We denote the size of some system X by $|X|$.

Theorem 16. *The trade-offs between $\mathscr{U}$-NPDA and DPDA, κ-NPDA and DPDA, as well as between $\mathscr{I}$-NPDA and DPDA are non-recursive.*

Proof. Assume in contrast to the assertion that one of the trade-offs is recursive. We turn to show that in this case the halting problem for Turing machines on empty tape would be decidable, a contradiction.

So, let M be some given Turing machine of size $c \geq 1$. Then a Turing machine M_c with unary input alphabet $\{a\}$ is uniformly constructed as follows. On input a^x, first M_c enumerates all the finitely many Turing machines whose size is c. Then it simulates all these finitely many Turing machines on empty tape by dovetailing for exactly x steps (or up to halting if a machine halts before x steps). Machine M_c accepts its input a^x by halting if at least one of the simulations halts exactly after step x. If not, M_c does not halt. We conclude that the language $L(M_c)$ accepted by M_c is finite. Moreover, the length of the longest word in $L(M_c)$ gives the latest time step at which a Turing machine of size c halts on empty tape.

The finiteness of $L(M_c)$ is easily proved by a proof Π_1. The length of Π_1 can be bounded from above by $\varrho_1(c)$, where ϱ_1 is a recursive function.

Next, from M_c an NPDA N accepting $\text{INVALC}(M_c)$ is constructed by Corollary 15. The corollary can be proved by a proof Π_2 whose length can be bounded from above by $\varrho_2(c)$, where ϱ_2 is a recursive function.

Since M_c accepts a finite language, $\text{INVALC}(M_c)$ is a co-finite and, thus, regular language. This fact is easily proved by a proof Π_3 whose length can be bounded from above by $\varrho_3(c)$, where ϱ_3 is a recursive function.

Since all regular languages belong to all Boolean kernels $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$, the proofs Π_1, Π_2, and Π_3 reveal that N is a $\mathscr{U}$-NPDA, a κ-NPDA, as well as an $\mathscr{I}$-NPDA. The total length of this proof, which is attached to N, can be bounded from above by $\varrho(c)$, where ϱ is a recursive function.

Recall that we assume in contrast to the assertion that the trade-off between the size of N (including the attached proof) and the size of some equivalent DPDA D is given by a recursive function f, that is $|D| \leq f(|N|)$. Then $f(|N|)$ can be computed from N, and all DPDAs whose size is at most $f(|N|)$ can be enumerated. Since the family of deterministic context-free languages is effectively closed under complementation, each DPDA P in the list can be replaced by a DPDA accepting the complement of $L(P)$. Since finiteness of deterministic context-free languages is decidable, each DPDA that accepts an infinite language can be removed from the new list. The decision of finiteness of deterministic context-free languages includes the computation of an upper bound for the length of the longest word in the language. So, an upper bound for the length of the longest word accepted by any of the remaining DPDAs in the list can be computed. Moreover, among the remaining DPDAs there must be one that accepts the finite complement of $L(D)$. The finite complement of $L(D)$ is the language $\text{VALC}(M_c)$. Clearly, the longest word in $\text{VALC}(M_c)$ is longer than the longest word in $L(M_c)$. As before, an upper bound of the length of this longest word can be computed from D. But recall that the length of the longest word in $L(M_c)$ gives the latest time step at which a Turing machine of size c halts on empty tape.

Finally, it suffices to simulate the given Turing machine M for this number of steps in order to decide whether it halts on empty tape or not. □

6 Untouched and Open Questions

We have started to study the properties of Boolean kernels of the family of context-free languages. Since little is known about kernels, many questions and problems remain open or untouched. Exemplarily, we mention some of them: (1) Is there a Boolean kernel $\kappa \in \gamma_{\text{BOOL}}(\text{CFL})$ that is closed under reversal or are *all* these kernels non-closed under reversal? (2) Further non-trivial closure properties of kernels are of natural interest. (3) A machine characterization of the complementation kernel of the context-free languages in terms of self-verifying pushdown automata is known [2]. Basically, the characterization is given by a machine for the underlying language family, where the acceptance condition is modified. Are there machine characterizations of other kernels? (4) In [5] an improved version of Earley's algorithm is adapted to context-free grammars which are extended by complement and intersection operators retaining cubic behavior. More generally, in [11] so-called conjunctive and Boolean grammars are studied. Can these techniques be applied to sub-families of context-free languages in order to find characterizations of kernels? (5) Alternative characterizations of kernels could be generators, where a generator is some smallest set of languages whose closure under certain operations is the kernel. Based on a generator, the languages in the kernel could be represented as expressions.

References

1. Bertsch, E., Nederhof, M.J.: Regular closure of deterministic languages. SIAM J. Comput. **29**, 81–102 (1999)
2. Fernau, H., Kutrib, M., Wendlandt, M.: Self-verifying pushdown automata. Fundam. Inform. **180**, 1–28 (2021)
3. Harrison, M.A.: Introduction to Formal Language Theory. Addison-Wesley, Boston (1978)
4. Hartmanis, J.: On the succinctness of different representations of languages. In: Maurer, H.A. (ed.) ICALP 1979. LNCS, vol. 71, pp. 282–288. Springer, Heidelberg (1979). https://doi.org/10.1007/3-540-09510-1_22
5. Heilbrunner, S., Schmitz, L.: An efficient recognizer for the Boolean closure of context-free languages. Theor. Comput. Sci. **80**, 53–75 (1991)
6. Holzer, M., Kutrib, M.: Descriptional complexity - an introductory survey. In: Martín-Vide, C. (ed.) Scientific Applications of Language Methods, pp. 1–58. Imperial College Press (2010)
7. Ilie, L., Păun, G., Rozenberg, G., Salomaa, A.: On strongly context-free languages. Discret. Appl. Math. **103**, 158–165 (2000)
8. Kutrib, M.: Kernels of sub-classes of context-free languages. In: Chatzigeorgiou, A., et al. (eds.) SOFSEM 2020. LNCS, vol. 12011, pp. 136–147. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-38919-2_12
9. Kutrib, M., Malcher, A.: Finite turns and the regular closure of linear context-free languages. Discret. Appl. Math. **155**, 2152–2164 (2007)

10. Kutrib, M., Malcher, A., Wotschke, D.: The Boolean closure of linear context-free languages. Acta Inform. **45**, 177–191 (2008)
11. Okhotin, A.: Boolean grammars. Inform. Comput. **194**, 19–48 (2004)
12. Wotschke, D.: Nondeterminism and Boolean operations in PDA's. J. Comput. Syst. Sci. **16**, 456–461 (1978)
13. Wotschke, D.: The Boolean closures of the deterministic and nondeterministic context-free languages. In: Brauer, W. (ed.) GI 1973. LNCS, vol. 1, pp. 113–121. Springer, Heidelberg (1973). https://doi.org/10.1007/3-540-06473-7_11
14. Wotschke, D.: Degree-languages: a new concept of acceptance. J. Comput. Syst. Sci. **14**, 187–209 (1977)