Semilinearity of Families of Languages

Oscar H. Ibarra[1(✉)] and Ian McQuillan[2]

[1] Department of Computer Science, University of California, Santa Barbara, CA 93106, USA
ibarra@cs.ucsb.edu

[2] Department of Computer Science, University of Saskatchewan, Saskatoon S7N 5A9, Canada
mcquillan@cs.usask.ca

Abstract. Techniques are developed for creating new and general language families of only semilinear languages, and for showing families only contain semilinear languages. It is shown that for language families $\mathcal{L}$ that are semilinear full trios, the smallest full AFL containing the languages obtained by intersecting languages in $\mathcal{L}$ with languages in NCM (where NCM is the family of languages accepted by NFAs augmented with reversal-bounded counters), is also semilinear. If these closure properties are effective, this also immediately implies decidability of membership, emptiness, and infiniteness for these general families. From the general techniques, new grammar systems are given that are extensions of well-known families of semilinear full trios, whereby it is implied that these extensions must only describe semilinear languages. This also implies positive decidability properties for the new systems. Some characterizations of the new families are also given.

1 Introduction

One-way nondeterministic reversal-bounded multicounter machines (NCM) operate like NFAs with λ transitions, where there are some number of stores that each can contain some non-negative integer. The transition function can detect whether each counter is zero or non-zero, and optionally increment or decrement each counter; however, there is a bound on the number of changes each counter can make between non-decreasing and non-increasing. These machines have been extensively studied in the literature, for example in [15], where it was shown that NCMs only accept semilinear languages (defined in Sect. 2). As the semilinear property is effective for NCM (in that, the proof consists of an algorithm for constructing a finite representation of the semilinear sets), this implies that NCMs have decidable membership, emptiness, and infiniteness properties, as emptiness and infiniteness can be decided easily on semilinear sets (and membership follows from emptiness by effective closure under intersection with regular languages).

The research of O. H. Ibarra was supported, in part, by NSF Grant CCF-1117708. The research of I. McQuillan was supported, in part, by the Natural Sciences and Engineering Research Council of Canada.

C. Câmpeanu (Ed.): CIAA 2018, LNCS 10977, pp. 211–222, 2018.
https://doi.org/10.1007/978-3-319-94812-6_18

NCM machines have been applied extensively in the literature, for example, to model checking and verification [16,17,21,22], often using the positive decidability properties of the family.

More general machine models have been studied with an unrestricted pushdown store augmented by some number of reversal-bounded counters (NPCM, [15]). Despite the unrestricted pushdown, the languages accepted are all semilinear, implying they have the same decidable properties. This family too has been applied to several verification problems [4,18], including model checking recursive programs with numeric data types [10], synchronization- and reversal-bounded analysis of multithreaded programs [8], for showing decidable properties of models of integer-manipulating programs with recursive parallelism [9], and for decidability of problems on commutativity [19]. In these papers, the positive decidability properties—the result of the semilinearity—plus the use of the main store (the pushdown), plus the counters, played a key role. Hence, (effective) semilinearity is a crucial property for families of languages.

The ability to augment a machine model with reversal-bounded counters and to only accept semilinear languages is not unique to pushdown automata; in [11], it was found that many classes of machines $\mathcal{M}$ accepting semilinear languages could be augmented with reversal-bounded counters, and the resulting family $\mathcal{M}_c$ would also only accept semilinear languages. This includes models such as Turing machines with a one-way read-only input tape and a finite-crossing[1] worktape. However, a precise formulation of which classes of machines this pertains to was not given.

Here, a precise formulation of families of languages that can be "augmented" with counters will be examined in terms of closure properties rather than machine models. This allows for application to families described by machine models, or grammatical models. It is shown that for any full trio (a family closed under homomorphism, inverse homomorphism, and intersection with regular languages) of semilinear languages $\mathcal{L}_0$, then the smallest full AFL $\mathcal{L}$ (a full trio also closed under union, concatenation, and Kleene-*) containing all languages obtained from intersecting a language in $\mathcal{L}_0$ with a language in NCM, must only contain semilinear languages. Furthermore, if the closure properties and semilinearity are effective in $\mathcal{L}_0$, this implies a decidable membership, emptiness, and infiniteness problem in $\mathcal{L}$. Hence, this provides a new method for creating general families of languages with positive decidability properties.

Several specific models are created by adding counters. For example, indexed grammars are a well-studied general grammatical model like context-free grammars except where nonterminals keep stacks of "indices". Although this system can generate non-semilinear languages, linear indexed grammars (indexed grammars with at most one nonterminal in the right hand side of every production) generate only semilinear languages [5]. Here, we define *linear indexed grammars with counters*, akin to linear indexed grammars, where every sentential form contains the usual sentential form, plus k counter values; each production operates as

[1] A worktape is finite-crossing if there is a bound on the number of times the boundary of all neighboring cells on the worktape are crossed.

usual and can also optionally increase each counter by some amount; and a terminal word can be generated only if it can be produced with all counter values equal. It is shown that the family of languages generated must be semilinear since it is contained in the smallest full AFL containing the intersection of linear indexed languages and NCM languages. A characterization is also shown: linear indexed grammars with counters generate exactly those languages obtained by intersecting a linear indexed language with an NCM and then applying a homomorphism. Furthermore, it is shown that right linear indexed grammars (where terminals only appear to the left of nonterminals in productions) with counters coincide exactly with the machine model NPCM. Therefore, linear indexed grammars with counters are a natural generalization of NPCM containing only semilinear languages. This model is generalized once again as follows: an indexed grammar is uncontrolled finite-index if, there is a value k such that, for every derivation in the grammar, there are at most k occurrences of nonterminals in every sentential form. It is known that every uncontrolled finite-index indexed grammar generates only semilinear languages [3,25]. It is shown here that uncontrolled finite-index indexed grammars with counters generate only semilinear languages, which is also a natural generalization of both linear indexed grammars with counters and NPCM. This immediately shows decidability of membership, emptiness, and infiniteness for this family.

Lastly, the closure property theoretic method of adding counters is found to often be more helpful than the machine model method of [11] in terms of determining whether the resulting family is semilinear, as here a machine model $\mathcal{M}$ is constructed such that the language family accepted by $\mathcal{M}$ is a semilinear full trio, but adding counters to the model to create $\mathcal{M}_c$ accepts non-semilinear languages. This implies from our earlier results, that $\mathcal{M}_c$ can accept languages that cannot be obtained by intersecting a language accepted by a machine in $\mathcal{M}$ with a NCM and then applying any of the full AFL properties.

This paper therefore contains useful new techniques for creating new language families, and for showing existing language families only contain semilinear languages, which can then be used to immediately obtain decidable emptiness, membership, and infiniteness problems. Such families can perhaps also be applied to various areas, such as to verification, similarly to the use of NPCM.

All proofs are omitted due to space constraints.

2 Preliminaries

In this section, preliminary background and notation is given.

Let $\mathbb{N}_0$ be the set of non-negative integers, and let $\mathbb{N}_0^k$ be the set of all k-tuples of non-negative integers. A set $Q \subseteq \mathbb{N}_0^k$ is *linear* if there exists vectors $\boldsymbol{v_0}, \boldsymbol{v_1}, \ldots, \boldsymbol{v_l} \in \mathbb{N}_0^k$ such that $Q = \{\boldsymbol{v_0} + i_1\boldsymbol{v_1} + \cdots + i_l\boldsymbol{v_l} \mid i_1, \ldots, i_l \in \mathbb{N}_0\}$. Here, $\boldsymbol{v_0}$ is called the *constant*, and $\boldsymbol{v_1}, \ldots, \boldsymbol{v_l}$ are called the *periods*. A set Q is called semilinear if it is a finite union of linear sets.

Introductory knowledge of formal language and automata theory is assumed such as nondeterministic finite automata (NFAs), pushdown automata (NPDAs),

Turing machines, and closure properties. [14]. An *alphabet* Σ is a finite set of symbols, a *word* w over Σ is a finite sequence of symbols from Σ, and Σ^* is the set of all words over Σ which includes the empty word λ. A *language* L over Σ is any $L \subseteq \Sigma^*$. The *complement* of a language $L \subseteq \Sigma^*$, denoted by $\overline{L}$, is $\Sigma^* - L$.

Given a word $w \in \Sigma^*$, the length of w is denoted by $|w|$. For $a \in \Sigma$, the number of a's in w is denoted by $|w|_a$. Given a word w over an alphabet $\Sigma = \{a_1, \ldots, a_k\}$, the Parikh map of w, $\psi(w) = (|w|_{a_1}, \ldots, |w|_{a_k})$, and the Parikh map of a language L is $\{\psi(w) \mid w \in L\}$. The commutative closure of a language L is the language $\text{comm}(L) = \{w \in \Sigma^* \mid \psi(w) = \psi(v), v \in L\}$. Two languages are *letter-equivalent* if $\psi(L_1) = \psi(L_2)$.

A language L is *semilinear* if $\psi(L)$ is a semilinear set. Equivalently, a language is semilinear if and only if it is letter-equivalent to some regular language [12]. A family of languages is semilinear if all languages in it are semilinear, and it is said that it is effectively semilinear if there is an algorithm to construct the constant and periods for each linear set from a representation of each language in the family. For example, it is well-known that all context-free languages are effectively semilinear [23].

Notation from AFL (abstract families of languages) theory is used from [6]. A *full trio* is any family of languages closed under homomorphism, inverse homomorphism, and intersection with regular languages. Furthermore, a *full AFL* is a full trio closed under union, concatenation, and Kleene-*. Given a language family $\mathcal{L}$, the smallest family containing $\mathcal{L}$ that is closed under arbitrary homomorphism is denoted by $\hat{\mathcal{H}}(\mathcal{L})$, the smallest full trio containing $\mathcal{L}$ is denoted by $\hat{\mathcal{M}}(\mathcal{L})$, and the smallest full AFL containing $\mathcal{L}$ is denoted by $\hat{\mathcal{F}}(\mathcal{L})$. Given families $\mathcal{L}_1$ and $\mathcal{L}_2$, let $\mathcal{L}_1 \wedge \mathcal{L}_2 = \{L_1 \cap L_2 \mid L_1 \in \mathcal{L}_1, L_2 \in \mathcal{L}_2\}$.

We will only define NCM and NPCM informally here, and refer to [15] for a formal definition. A one-way nondeterministic counter machine can be defined equivalently to a one-way nondeterministic pushdown automaton [14] with only a bottom-of-pushdown marker plus one other symbol. Hence, the machine can add to the counter (by pushing), subtract from the counter (by popping), and can detect emptiness and non-emptiness of the pushdown. A k-counter machine has k independent counters. A k-counter machine M is l-reversal-bounded, if M makes at most l changes between non-decreasing and non-increasing of each counter in every accepting computation. Let NCM be the class of one-way nondeterministic l-reversal-bounded k-counter machines, for some k, l (DCM for deterministic machines). Let NPCM be the class of machines with one unrestricted pushdown plus some number of reversal-bounded counters. By a slight abuse of notation, we also use these names for the family of languages they accept.

3 Full AFLs Containing Counter Languages

This section will start by showing that for every semilinear full trio $\mathcal{L}$, the smallest full AFL containing $\mathcal{L} \wedge \text{NCM}$ is a semilinear full AFL. First, the following intermediate result is required.

Lemma 1. *If $\mathcal{L}$ is a semilinear full trio, then $\hat{\mathcal{M}}(\mathcal{L} \wedge \mathsf{NCM}) = \hat{\mathcal{H}}(\mathcal{L} \wedge \mathsf{NCM})$ is a semilinear full trio.*

The next result is relatively straightforward from results in [6,7], however we have not seen it explicitly stated as we have done. From Corollary 2, Sect. 3.4 of [6], for any full trio $\mathcal{L}$, the smallest full AFL containing $\mathcal{L}$ is the substitution of the regular languages into $\mathcal{L}$. And from [7], the substitution closure of one semilinear family into another is semilinear. Therefore, we obtain:

Lemma 2. *If $\mathcal{L}$ is a semilinear full trio, then the smallest full AFL containing $\mathcal{L}$ is semilinear.*

From these, it is immediate that for semilinear full trios $\mathcal{L}$, the smallest full AFL containing intersections of languages in $\mathcal{L}$ with NCM is semilinear.

Theorem 3. *If $\mathcal{L}$ is a semilinear full trio $\mathcal{L}$, then $\hat{\mathcal{F}}(\mathcal{L} \wedge \mathsf{NCM})$ is semilinear.*

It is worth noting that this procedure can be iterated, as therefore $\hat{\mathcal{F}}(\hat{\mathcal{F}}(\mathcal{L} \wedge \mathsf{NCM}) \wedge \mathsf{NCM})$ must also be a semilinear full AFL, etc. for additional levels, but it is not clear whether this can increase the capacity or not.

Many acceptors and grammar systems are known to be semilinear full trios, such as finite-index $\mathsf{ET0L}$ systems [24], indexed grammars with a bound on the number of variables appearing in every sentential form (called uncontrolled finite-index) [3], multi-push-down machines (which have k pushdowns that can simultaneously be written to, but they can only pop from the first non-empty pushdown) [2], a Turing machine variant with one finite-crossing worktape [11], and pushdown machines that can flip their pushdown up to k times [13].

Corollary 4. *Let $\mathcal{L}$ be any of the following families:*

- *languages generated by context-free grammars,*
- *languages generated by finite-index ETOL,*
- *languages generated by uncontrolled finite-index indexed languages,*
- *languages accepted by one-way multi-push-down machine languages,*
- *languages accepted by one-way read-only input nondeterministic Turing machines with a two-way finite-crossing read/write worktape,*
- *languages accepted by one-way k-flip pushdown automata.*

Then the smallest full AFL containing $\mathcal{L} \wedge \mathsf{NCM}$ is a semilinear full AFL.

A simplified analogue to this result is known for certain types of machines [11], although the new result here is defined entirely using closure properties rather than machines. Furthermore, the results in [11] do not allow Kleene-* type closure as part of the full AFL properties. For the machine models $\mathcal{M}$ above, it is an easy exercise to show that augmenting them with reversal-bounded counters to produce $\mathcal{M}_c$, the languages accepted by $\mathcal{M}_c$ are a subset of the smallest full AFL containing intersections of languages in $\mathcal{M}$ with NCM. Hence, these models augmented by counters only accept semilinear languages. Similarly, this type of technique also works for grammar systems, as seen in Sect. 5.

In addition, in [7], it was shown that if $\mathcal{L}$ is a semilinear family, then the smallest AFL containing the commutative closure of $\mathcal{L}$ is a semilinear AFL. It is known that the commutative closure of every semilinear language is in NCM [19], and we know now that if we have a semilinear full trio $\mathcal{L}$, then the smallest full AFL containing $\mathcal{L}$ is also semilinear. So, we obtain an alternate proof that is an immediate corollary since we know that the smallest full AFL containing NCM is a semilinear full AFL.

For any semilinear full trio $\mathcal{L}$ where the semilinearity and the intersection with regular language properties are effective, the membership and emptiness problems in $\mathcal{L}$ are decidable. Indeed, to decide emptiness, it suffices to check if the semilinear set is empty. And to decide if a word w is in L, one constructs the language $L \cap \{w\}$, then emptiness is decided.

Corollary 5. *For any semilinear full trio $\mathcal{L}$ where the semilinearity and intersection with regular language properties are effective, then the membership, emptiness, and infiniteness problems are decidable for languages in $\hat{\mathcal{F}}(\mathcal{L} \wedge \mathsf{NCM})$. In these cases, $\hat{\mathcal{F}}(\mathcal{L} \wedge \mathsf{NCM})$ are a strict subset of the recursive languages.*

As membership is decidable, the family must only contain recursive languages, and the inclusion must be strict as the recursive languages are not closed under homomorphism.

The next property on commutative closure also follows.

Proposition 6. *Let $\mathcal{L}$ be a semilinear full trio, where these properties are effective. Then, the problem, for $L_1, L_2 \in \hat{\mathcal{F}}(\mathcal{L} \wedge \mathsf{NCM})$ is $L_1 \subseteq \mathrm{comm}(L_2)$, is decidable. Furthermore, the problem, is $L_1 \cap \mathrm{comm}(L_2) = \emptyset$ is decidable.*

Next, we provide an interesting decomposition theorem of semilinear languages into linear parts. Consider any semilinear language L, where its Parikh image is a finite union of linear sets $A_1, \ldots, A_k$, and the constant and periods for each linear set can be constructed. Then we can effectively create languages in perhaps another semilinear full trio separately accepting those words in $L_i = \{w \in L \mid \psi(w) \in A_i\}$, for each $1 \leq i \leq k$.

Corollary 7. *Let $\mathcal{L}$ be a semilinear full trio, where semilinearity is effective. Then, given $L \in \mathcal{L}$, we can determine that the Parikh map of L is $A = A_1 \cup \cdots \cup A_k$, $A_1, \ldots, A_k$ are linear sets, and we can effectively construct languages $L_1, \ldots, L_k$ in the semilinear full trio $\hat{\mathcal{M}}(\mathcal{L} \wedge \mathsf{NCM})$ such that $L_i = \{w \in L \mid \psi(w) \in A_i\}$.*

Proof. Since semilinearity is effective, we can construct a representation of linear sets $A_1, \ldots, A_k$. An NCM M_i can be created to accept $\psi^{-1}(A_i)$, for each i, $1 \leq i \leq k$. Then, $L_i = L \cap L(M_i) \in \hat{\mathcal{M}}(\mathcal{L} \wedge \mathsf{NCM})$, for each i, $1 \leq i \leq k$. □

4 Application to General Multi-store Machine Models

In [6], a generalized type of multitape automata was studied, called multitape abstract families of automata (multitape AFAs). We will not define the notation

used there, but in Theorem 4.6.1 (and Exercise 4.6.3), it is shown that if we have two types of automata $\mathcal{M}_1$ and $\mathcal{M}_2$ (defined using the AFA formalism), accepting language families $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively, then the languages accepted by automata combining together the stores of $\mathcal{M}_1$ and $\mathcal{M}_2$, accepts exactly the family $\hat{\mathcal{H}}(\mathcal{L}_1 \wedge \mathcal{L}_2)$. This is shown for machines accepting full AFLs in Theorem 4.6.1 of [6], and for union-closed full trios mentioned in Exercise 4.6.3. We will show that this is tightly coupled with this precise definition of AFAs, as we will define a simple type of multitape automata where this is not the case, but each type still satisfies the same closure properties. This result uses the characterization of Theorem 3.

A checking stack automaton (NCSA) M is a one-way NFA with a store tape, called a stack. At each move, M pushes a string (possibly λ) on the stack, but M cannot pop. And, M can enter and read from the inside of the stack in two-way read-only fashion. But once the machine enters the stack, it can no longer change the contents. The checking stack automaton is said to be *restricted* (or *no-read* using the terminology of [20]), if it does not read from the inside of the stack until the end of the input. We denote by RNCSA the family of machines, as well as the family of languages described by the machines above, with RDCSA being the deterministic version. Let RNCSA_c (RDCSA_c) be the family of machines and languages in RNCSA (RDCSA) augmented with reversal-bounded counters. A preliminary investigation of RNCSA_c and RDCSA_c was done in [20].

Here, we will show the following:

1. RNCSA is a full trio of semilinear languages,
2. $\hat{\mathcal{F}}(\mathsf{RNCSA} \wedge \mathsf{NCM})$ is a semilinear full AFL,
3. every language in $\mathsf{RNCSA} \wedge \mathsf{NCM}$ is accepted by some machine in RNCSA_c,
4. there are non-semilinear languages accepted by machines in RNCSA_c.

Therefore, RNCSA_c contains some languages not in the smallest full AFL containing $\mathsf{RNCSA} \wedge \mathsf{NCM}$, and the multitape automata and results from [6,11] do not apply to this type of automata.

Proposition 8. *RNCSA accepts exactly the regular languages, which is a full trio of semilinear languages.*

From Theorem 3, the following is true:

Corollary 9. $\hat{\mathcal{F}}(\mathit{RNCSA} \wedge \mathsf{NCM})$ *is a semilinear full AFL.*

Since RNCSA accepts the regular languages, and NCM is closed under intersection with regular languages, the following is true:

Proposition 10. $\mathit{RNCSA} \wedge \mathsf{NCM} = \mathsf{NCM} \subseteq \mathit{RNCSA}_c$.

Proposition 11. *The non-semilinear* $L = \{a^i b^j \mid i, j \geq 1, j \text{ is divisible by } i\}$ *can be accepted by an* RDCSA_c M *with one counter that makes only one reversal.*

It is concluded that RNCSA_c contains some languages not in $\mathsf{RNCSA} \wedge \mathsf{NCM} = \mathsf{NCM}$, since NCM is semilinear [15]. Moreover, $\hat{\mathcal{F}}(\mathsf{RNCSA} \wedge \mathsf{NCM})$ is semilinear

as well, so it does not contain all languages of RNCSA_c. Then it is clear that combining together the stores of RNCSA and NCM accepts significantly more than $\hat{\mathcal{H}}(\mathsf{RNCSA} \wedge \mathsf{NCM})$ as is the case for multitape AFA [6]. The reason for the discrepancy between this result and Ginsburg's result is that the definition of multitape AFA allows for reading the input while performing instructions (like operating in two-way read-only mode in the stack). In contrast, RNCSA does not allow this behavior. And if this behavior is added into the definition, the full capability of checking stack automata is achieved which accepts non-semilinear languages, and not regular languages.

A similar analysis can be done using the method developed in [11] for augmenting the machine models with counters. Let $\mathcal{M}$ be a family of one-way acceptors with some type of store structure X. For example, if the storage X is a pushdown stack, then $\mathcal{M}$ is the family of nondeterministic pushdown automata (NPDAs). Let the machines in $\mathcal{M}$ be augmented with reversal-bounded counters, and call the resulting family $\mathcal{M}_c$. In [11], the following was shown for many families $\mathcal{M}$:

(*) If $\mathcal{M}$ is a semilinear family (i.e., the languages accepted by the machines in $\mathcal{M}$ have semilinear Parikh map), then $\mathcal{M}_c$ is also a semilinear family.

It was not clear in [11] whether the result above is true for all types of one-way acceptors, in general. However, the family RNCSA is semilinear (Proposition 8), but RDCSA_c is not semilinear (Proposition 11).

5 Applications to Indexed Grammars with Counters

In this section, we describe some new types of grammars obtained from existing grammars generating a semilinear language family $\mathcal{L}$, by adding counters. The languages generated by these new grammars are then shown to be contained in $\hat{\mathcal{F}}(\mathcal{L} \wedge \mathsf{NCM})$, and by an application of Theorem 3, are all semilinear with positive decidability properties.

We need the definition of an indexed grammar introduced in [1] by following the notation of [14], Sect. 14.3.

Definition 12. *An indexed grammar is a 5-tuple $G = (V, \Sigma, I, P, S)$, where V, Σ, I are finite pairwise disjoint sets: the set of nonterminals, terminals, and indices, respectively, S is the start nonterminal, and P is a finite set of productions, each of the form either*

$$\textbf{(1)}\ A \to \nu, \quad \textbf{(2)}\ A \to Bf, \quad \textit{or} \quad \textbf{(3)}\ Af \to \nu,$$

where $A, B \in V$, $f \in I$ and $\nu \in (V \cup \Sigma)^*$.

Let ν be an arbitrary sentential form of G, which is of the form

$$\nu = u_1 A_1 \alpha_1 u_2 A_2 \alpha_2 \cdots u_k A_k \alpha_k u_{k+1},$$

where $A_i \in V, \alpha_i \in I^*, u_i \in \Sigma^*, 1 \leq i \leq k, u_{k+1} \in \Sigma^*$. For a sentential form $\nu' \in (VI^* \cup \Sigma)^*$, we write $\nu \Rightarrow_G \nu'$ if one of the following three conditions holds:

1. There exists a production in P of the form (1) $A \to w_1 C_1 \cdots w_\ell C_\ell w_{\ell+1}$, $C_j \in V, w_j \in \Sigma^*$, and there exists i with $1 \leq i \leq k$, $A_i = A$ and
$$\nu' = u_1 A_1 \alpha_1 \cdots u_i (w_1 C_1 \alpha_i \cdots w_\ell C_\ell \alpha_i w_{\ell+1}) u_{i+1} A_{i+1} \alpha_{i+1} \cdots u_k A_k \alpha_k u_{k+1}.$$
2. There exists a production in P of the form (2) $A \to Bf$ and there exists i, $1 \leq i \leq k$, $A_i = A$ and $\nu' = u_1 A_1 \alpha_1 \cdots u_i (B f \alpha_i) u_{i+1} A_{i+1} \alpha_{i+1} \cdots u_k A_k \alpha_k u_{k+1}$.
3. There exists a production in P of the form (3) $Af \to w_1 C_1 \cdots w_\ell C_\ell w_{\ell+1}$, $C_j \in V, w_j \in \Sigma^*$, and an i, $1 \leq i \leq k$, $A_i = A$, $\alpha_i = f\alpha_i', \alpha_i' \in I^*$, with $\nu' = u_1 A_1 \alpha_1 \cdots u_i (w_1 C_1 \alpha_i' \cdots w_\ell C_\ell \alpha_i' w_{\ell+1}) u_{i+1} A_{i+1} \alpha_{i+1} \cdots u_k A_k \alpha_k u_{k+1}$.

Then, $\Rightarrow_G^*$ denotes the reflexive and transitive closure of $\Rightarrow_G$. The language $L(G)$ generated by G is the set $L(G) = \{u \in \Sigma^* \mid S \Rightarrow_G^* u\}$.

This type of grammar can be generalized to include counters as follows:

Definition 13. *An* indexed grammar with k counters *is defined as in indexed grammars, except where rules (1), (2), (3) above are modified so that a rule* $\alpha \to \beta$ *now becomes:*
$$\alpha \to (\beta, c_1, \ldots, c_k), \tag{1}$$
where $c_i \geq 0$, $1 \leq i \leq k$. *Sentential forms are of the form* $(\nu, n_1, \ldots, n_k)$, *and* $\Rightarrow_G$ *operates as do indexed grammars on* ν, *and for a production in Eq. 1, adds* c_i *to* n_i, *for* $1 \leq i \leq k$. *The language generated by* G *with terminal alphabet* Σ *and start nonterminal* S *is,* $L(G) = \{w \mid w \in \Sigma^*, (S, 0, \ldots, 0) \Rightarrow_G^* (w, n_1, \ldots, n_k), n_1 = \cdots = n_k\}$.

Given an indexed grammar with counters, the underlying grammar *is the indexed grammar obtained by removing the counter components from productions.*

Although indexed grammars generate non-semilinear languages, restrictions will be studied that only generate semilinear languages.

An indexed grammar G is *linear* [5] if the right side of every production of G has at most one variable. Furthermore, G is *right linear* if it is linear, and terminals can only appear to the left of a nonterminal in productions. Let L-IND be the family of languages generated by linear indexed grammars, and let RL-IND be the family of languages generated by right linear indexed grammars.

Similarly, indexed grammars with counters can be restricted to be linear. An indexed grammar with k-counters is said to be *linear indexed* (resp. *right linear*) with k counters, if the underlying grammar is linear (resp. right linear). Let L-IND_c (resp. RL-IND_c) be the family of languages generated by linear (resp. right linear) indexed grammars with counters.

Example 14. Consider the language $L = \{v\$w \mid v, w \in \{a, b, c\}^*, |v|_a = |v|_b = |v|_c, |w|_a = |w|_b = |w|_c\}$ which can be generated by a linear indexed grammar with counters $G = (V, \Sigma, I, P, S)$ where P contains

$S \to (S, 1, 1, 1, 0, 0, 0) \mid (S, 0, 0, 0, 1, 1, 1) \mid (T, 0, 0, 0, 0, 0, 0)$

$T \to (aT, 1, 0, 0, 0, 0, 0) \mid (bT, 0, 1, 0, 0, 0, 0) \mid (cT, 0, 0, 1, 0, 0, 0) \mid (\$R, 0, 0, 0, 0, 0, 0)$

$R \to (aR, 0, 0, 0, 1, 0, 0) \mid (bR, 0, 0, 0, 0, 1, 0) \mid (cR, 0, 0, 0, 0, 0, 1) \mid (\lambda, 0, 0, 0, 0, 0, 0)$.

This language cannot be generated by a linear indexed grammar [3].

The following is a characterization of languages generated by these grammars.

Proposition 15. *$L \in$ L-IND$_c$ if and only if there is a homomorphism h, $L_1 \in$ L-IND, and $L_2 \in$ NCM such that $L = h(L_1 \cap L_2)$.*

Implied from the above result and Theorem 3 and since L-IND is an effectively semilinear trio [5] is that L-IND$_c \subseteq \hat{\mathcal{F}}$(L-IND $\wedge$ NCM), and therefore L-IND$_c$ is effectively semilinear.

Corollary 16. *The languages generated by linear indexed grammar with counters are effectively semilinear, with decidable emptiness, membership, and infiniteness problems.*

Next, a machine model characterization of right linear indexed grammars with counters will be provided. Recall that an NPCM is a pushdown automaton augmented by reversal-bounded counters. The proof uses the fact that every context-free language can be generated by a right-linear indexed grammar [5].

Theorem 17. *RL-IND$_c$ = NPCM.*

We conjecture that the family of languages generated by right-linear indexed grammars with counters (the family of NPCM languages) is properly contained in the family of languages generated by linear indexed grammars with counters. Candidate witness languages are $L = \{w\$w \mid w \in \{a, b, c\}^*, |w|_a + |w|_b = |w|_c\}$ and $L' = \{w\$w \mid w \in \{a, b\}^*\}$. It is known that L' is generated by a linear indexed grammar [5], and hence L can be generated by such a grammar with two counters. But, both L' and L seem unlikely to be accepted by any NPCM. Therefore, indexed grammars with counters form quite a general semilinear family as it seems likely to be more general than NPCM.

Next, another subfamily of indexed languages is studied that are even more expressive than linear indexed grammars but only generate semilinear languages.

An indexed grammar $G = (V, \Sigma, I, P, S)$ is said to be *uncontrolled* index-r if, every sentential form in every successful derivation has at most r nonterminals. G is uncontrolled finite-index if G is uncontrolled index-r, for some r. Let U-IND be the languages generated by uncontrolled finite-index indexed grammars.

Uncontrolled finite-index indexed grammars have also been studied under the name of breadth-bounded indexed grammars in [3,25], where it was shown that the languages generated by these grammars are a semilinear full trio.

This concept can then be carried over to indexed grammars with counters.

Definition 18. *An indexed grammar with k-counters is uncontrolled index-r (resp. uncontrolled finite-index) if the underlying grammar is uncontrolled index-r (resp. uncontrolled finite-index). Let U-IND$_c$ be the languages generated by uncontrolled finite-index indexed grammar with k-counters, for some k.*

One can easily verify that Proposition 15 also applies to uncontrolled finite-index indexed grammars with counters. Hence, we have:

Proposition 19. *$L \in$ U-IND$_c$ if and only if there is a homomorphism h, $L_1 \in$ U-IND, $L_2 \in$ NCM such that $L = h(L_1 \cap L_2)$.*

Implied from the above Proposition and Theorem 3 also is that these new languages are all semilinear.

Corollary 20. *U-IND$_c$ is effectively semilinear, with decidable emptiness, membership, and infiniteness problems.*

Hence, RL-IND$_c \subseteq$ L-IND$_c \subseteq$ U-IND$_c$. We conjecture that both containments are strict; the first was discussed previously, and the second is likely true since L-IND $\subsetneq$ U-IND [3]. Hence, U-IND$_c$ forms quite a general semilinear family, containing NPCM with positive decidability properties.

References

1. Aho, A.V.: Indexed grammars–an extension of context-free grammars. J. ACM **15**(4), 647–671 (1968)
2. Breveglieri, L., Cherubini, A., Citrini, C., Reghizzi, S.: Multi-push-down languages and grammars. Int. J. Found. Comput. Sci. **7**(3), 253–291 (1996)
3. D'Alessandro, F., Ibarra, O.H., McQuillan, I.: On finite-index indexed grammars and their restrictions. In: Drewes, F., Martín-Vide, C., Truthe, B. (eds.) LATA 2017. LNCS, vol. 10168, pp. 287–298. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53733-7_21
4. Dang, Z., Ibarra, O.H., Bultan, T., Kemmerer, R.A., Su, J.: Binary reachability analysis of discrete pushdown timed automata. In: Emerson, E.A., Sistla, A.P. (eds.) CAV 2000. LNCS, vol. 1855, pp. 69–84. Springer, Heidelberg (2000). https://doi.org/10.1007/10722167_9
5. Duske, J., Parchmann, R.: Linear indexed languages. Theoret. Comput. Sci. **32**(1–2), 47–60 (1984)
6. Ginsburg, S.: Algebraic and Automata-Theoretic Properties of Formal Languages. North-Holland Publishing Company, Amsterdam (1975)
7. Ginsburg, S., Spanier, E.H.: AFL with the semilinear property. J. Comput. Syst. Sci. **5**(4), 365–396 (1971)
8. Hague, M., Lin, A.W.: Synchronisation- and reversal-bounded analysis of multithreaded programs with counters. In: Madhusudan, P., Seshia, S.A. (eds.) CAV 2012. LNCS, vol. 7358, pp. 260–276. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31424-7_22
9. Hague, M., Lin, A.W.: Decidable models of integer-manipulating programs with recursive parallelism. In: Larsen, K.G., Potapov, I., Srba, J. (eds.) RP 2016. LNCS, vol. 9899, pp. 148–162. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45994-3_11
10. Hague, M., Lin, A.W.: Model checking recursive programs with numeric data types. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 743–759. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22110-1_60
11. Harju, T., Ibarra, O.H., Karhumäki, J., Salomaa, A.: Some decision problems concerning semilinearity and commutation. J. Comput. Syst. Sci. **65**(2), 278–294 (2002)
12. Harrison, M.: Introduction to Formal Language Theory. Addison-Wesley Series in Computer Science. Addison-Wesley Pub. Co., Boston (1978)

13. Holzer, M., Kutrib, M.: Flip-pushdown automata: nondeterminism is better than determinism. In: Ésik, Z., Fülöp, Z. (eds.) DLT 2003. LNCS, vol. 2710, pp. 361–372. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45007-6_29
14. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading (1979)
15. Ibarra, O.H.: Reversal-bounded multicounter machines and their decision problems. J. ACM **25**(1), 116–133 (1978)
16. Ibarra, O.H., Bultan, T., Su, J.: Reachability analysis for some models of infinite-state transition systems. In: Palamidessi, C. (ed.) CONCUR 2000. LNCS, vol. 1877, pp. 183–198. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44618-4_15
17. Ibarra, O.H., Bultan, T., Su, J.: On reachability and safety in infinite-state systems. Int. J. Found. Comput. Sci. **12**(6), 821–836 (2001)
18. Ibarra, O.H., Dang, Z.: Eliminating the storage tape in reachability constructions. Theoret. Comput. Sci. **299**(1–3), 687–706 (2003)
19. Ibarra, O.H., McQuillan, I.: The effect of end-markers on counter machines and commutativity. Theoret. Comput. Sci. **627**, 71–81 (2016)
20. Ibarra, O.H., McQuillan, I.: Variations of checking stack automata: obtaining unexpected decidability properties. In: Charlier, É., Leroy, J., Rigo, M. (eds.) DLT 2017. LNCS, vol. 10396, pp. 235–246. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62809-7_17
21. Ibarra, O.H., Su, J., Dang, Z., Bultan, T., Kemmerer, R.: Counter machines and verification problems. Theoret. Comput. Sci. **289**(1), 165–189 (2002)
22. Ibarra, O.H., Su, J., Dang, Z., Bultan, T., Kemmerer, R.: Counter machines: decidable properties and applications to verification problems. In: Nielsen, M., Rovan, B. (eds.) MFCS 2000. LNCS, vol. 1893, pp. 426–435. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44612-5_38
23. Parikh, R.: On context-free languages. J. ACM **13**(4), 570–581 (1966)
24. Rozenberg, G., Vermeir, D.: On ET0L systems of finite index. Inf. Control **38**, 103–133 (1978)
25. Zetzsche, G.: An approach to computing downward closures. In: Halldórsson, M.M., Iwama, K., Kobayashi, N., Speckmann, B. (eds.) ICALP 2015. LNCS, vol. 9135, pp. 440–451. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47666-6_35