ON THE EXPRESSIVENESS OF PARIKH AUTOMATA AND RELATED MODELS

Michaël Cadilhac[1], Alain Finkel[2], and Pierre McKenzie[1]

[1] DIRO, Université de Montréal
{cadilhac, mckenzie}@iro.umontreal.ca
[2] LSV, CNRS & École Normale Supérieure de Cachan
finkel@lsv.ens-cachan.fr

Abstract

The Parikh finite word automaton (PA) was introduced and studied by Klaedtke and Rueß [16]. Natural variants of the PA arise from viewing a PA equivalently as an automaton that keeps a count of its transitions and semilinearly constrains their numbers. Here we adopt this view and define the affine PA (APA), that extends the PA by having each transition induce an affine transformation on the PA registers, and the PA on letters (LPA), that restricts the PA by forcing any two transitions on same letter to affect the registers equally. Then we report on the expressiveness, closure, and decidability properties of such PA variants. We note that deterministic PA are strictly weaker than deterministic reversal-bounded counter machines. We develop pumping-style lemmas and identify an explicit PA language recognized by no deterministic PA. Our findings and the resulting overall picture are tabulated in our concluding section.

1. Introduction

Adding features to finite automata in order to capture situations beyond regularity has been fruitful to many areas of research, in particular model checking and complexity theory below NC^2 (e.g., [17, 21]). One such finite automaton extension is the *Parikh automaton* (PA): A PA [16] is a pair (A, C) where C is a semilinear subset of $\mathbb{N}^d$ and A is a finite automaton over $(\Sigma \times D)$ for Σ a finite alphabet and D a finite subset of $\mathbb{N}^d$. The PA accepts the word $w_1 \cdots w_n \in \Sigma^*$ if A accepts a word $(w_1, \overline{v_1}) \cdots (w_n, \overline{v_n})$ such that $\sum \overline{v_i} \in C$. Klaedtke and Rueß used PA to characterize an extension of (existential) monadic second-order logic in which the cardinality of sets expressed by second-order variables is available.

Here we carry the study of Parikh automata a little further. First we introduce related models of independent interest, each involving a finite automaton A and a *constraint set* C of vectors. (The main text has formal definitions.) (1) *Constrained automata* (CA) are defined to accept a word $w \in \Sigma^*$ iff the Parikh image of some accepting run of A on w (i.e., the vector recording the number of occurrences of each transition along the run) belongs to C. (2) *Affine Parikh automata* (APA) generalize PA by allowing each transition to perform a linear transformation on the d-tuple of PA registers prior to adding a new vector; an APA accepts a word w iff some accepting run of A on w maps the all-zero vector to a vector in C. (3) *Parikh automata on* *letters* (LPA) restrict PA by imposing the condition that any transition on $(a, \overline{u}) \in (\Sigma \times D)$ and any transition on $(b, \overline{v}) \in (\Sigma \times D)$ must satisfy $\overline{u} = \overline{v}$ when $a = b$.

Then our main observations are the following:

- CA and deterministic CA respectively capture the class $\mathcal{L}_{\text{PA}}$ of PA languages and the class $\mathcal{L}_{\text{DetPA}}$ of deterministic PA languages.
- The language $\{a, b\}^* \cdot \{a^n \# a^n \mid n \in \mathbb{N}\}$ belongs to $\mathcal{L}_{\text{PA}} \setminus \mathcal{L}_{\text{DetPA}}$; these two classes were only proved different in [16].
- APA and deterministic APA over $\mathbb{Q}$ are no more powerful than the same models over $\mathbb{N}$.
- APA express more languages than PA, and only context-sensitive languages; moreover the emptiness problem for deterministic APA is already undecidable.
- Languages of LPA are equivalent to regular languages with a constraint on the Parikh image of their words.
- Refining [16] slightly, we compare our models with the reversal-bounded counter machines (RBCM) defined by Ibarra [12], and show that $\mathcal{L}_{\text{DetPA}}$ is a strict subset of the languages expressed by deterministic RBCM.
- Further expressiveness properties, closure properties, decidability properties and comparisons between the above models are derived. The overall resulting picture is summarized in tabular form in Section 6.

2. Preliminaries

We write $\mathbb{Z}$ for the integers, $\mathbb{N}$ for the nonnegative integers, $\mathbb{N}^+$ for $\mathbb{N} \setminus \{0\}$, $\mathbb{Q}$ for the rational numbers, and $\mathbb{Q}^+$ for the strictly positive rational numbers. We use $\mathbb{K}$ to denote either $\mathbb{N}$ or $\mathbb{Q}$. Let $d, d' \in \mathbb{N}^+$. Vectors in $\mathbb{K}^d$ are noted with a bar on top, e.g., $\overline{v}$ whose elements are $v_1, \ldots, v_d$. For $C \subseteq \mathbb{K}^d$ and $D \subseteq \mathbb{K}^{d'}$, we write $C.D$ for the set of vectors in $\mathbb{K}^{d+d'}$ which are the concatenation of a vector of C and a vector of D. We write $\overline{0} \in \{0\}^d$ for the all-zero vector, and $\overline{e_i} \in \{0, 1\}^d$ for the vector having a 1 only in position i. We view $\mathbb{K}^d$ as the additive monoid $(\mathbb{K}^d, +)$. For a monoid $(M, \cdot)$ and $S \subseteq M$, we write S^* for the monoid generated by S, i.e., the smallest submonoid of $(M, \cdot)$ containing S. A subset E of $\mathbb{K}^d$ is $\mathbb{K}$-*definable* if it is expressible as a first order formula which uses the function symbols $+$, λ_e with $e \in \mathbb{K}$ corresponding to the scalar multiplication, and the order $<$. More precisely, a subset E of $\mathbb{K}^d$ is $\mathbb{K}$-definable iff there is such a formula with d free variables, with $(x_1, \ldots, x_d) \in E \Leftrightarrow \mathbb{K} \models \phi(x_1, \ldots, x_d)$. Let us remark that $\mathbb{N}$-definable sets are the Presburger-definable sets and they coincide with the *semilinear sets* [9], i.e., finite unions of sets of the form $\{\overline{a_0} + k_1\overline{a_1} + \cdots + k_n\overline{a_n} \mid (\forall i)[k_i \in \mathbb{N}]\}$ for some $\overline{a_i}$'s in $\mathbb{N}^d$. Moreover, $\mathbb{Q}$-definable sets are the semialgebraic sets defined using affine functions[1] [6, Corollary I.7.8].

Let $\Sigma = \{a_1, \ldots, a_n\}$ be an (ordered) alphabet, and write ε for the empty word. The *Parikh image* is the morphism $\Phi \colon \Sigma^* \to \mathbb{N}^n$ defined by $\Phi(a_i) = \overline{e_i}$, for $1 \le i \le n$. A language $L \subseteq \Sigma^*$ is said to be *semilinear* if $\Phi(L) = \{\Phi(w) \mid w \in L\}$ is semilinear. The *commutative closure* of a language L is defined as the language $c(L) = \{w \mid \Phi(w) \in \Phi(L)\}$. A language $L \subseteq \Sigma^*$ is said

[1] Semialgrebraic sets defined using affine functions are sometimes also called semilinear (e.g., [6]). In this paper, we use "semilinear" only for $\mathbb{N}$-definable sets.

to be *bounded* if there exist $n > 0$ and $w_1, \ldots, w_n \in \Sigma^+$ such that $L \subseteq w_1^* \cdots w_n^*$. Two words $u, v \in \Sigma^*$ are *equivalent by the Nerode relation* (w.r.t. L), if for all $w \in \Sigma^*$, $uw \in L \Leftrightarrow vw \in L$. We then write $u \equiv_L v$ (or $u \equiv v$ when L is understood), and write $[u]_L$ for the equivalence class of u w.r.t. the Nerode relation.

We then fix our notation about automata. An automaton is a quintuple $A = (Q, \Sigma, \delta, q_0, F)$ where Q is the finite set of states, Σ is an alphabet, $\delta \subseteq Q \times \Sigma \times Q$ is the set of transitions, $q_0 \in Q$ is the initial state and $F \subseteq Q$ are the final states. For a transition $t \in \delta$, where $t = (q, a, q')$, we define $\mathsf{From}(t) = q$ and $\mathsf{To}(t) = q'$. Moreover, we define $\mu_A \colon \delta^* \to \Sigma^*$ to be the morphism defined by $\mu_A(t) = a$, and we write μ when A is clear from the context. A *path* on A is a word $\pi = t_1 \cdots t_n \in \delta^*$ such that $\mathsf{To}(t_i) = \mathsf{From}(t_{i+1})$ for $1 \leq i < n$; we extend From and To to paths, letting $\mathsf{From}(\pi) = \mathsf{From}(t_1)$ and $\mathsf{To}(\pi) = \mathsf{To}(t_n)$. We say that $\mu(\pi)$ is the *label* of π. A path π is said to be *accepting* if $\mathsf{From}(\pi) = q_0$ and $\mathsf{To}(\pi) \in F$; we let $\mathsf{Run}(A)$ be the language over δ of accepting paths on A. We then define $L(A)$, the *language* of A, as the labels of the accepting paths.

3. Parikh automata

The following notations will be used in defining Parikh finite word automata (PA) formally. Let Σ be an alphabet, $d \in \mathbb{N}^+$, and D a finite subset of $\mathbb{N}^d$. Following [16], the monoid morphism from $(\Sigma \times D)^*$ to Σ^* defined by $(a, \overline{v}) \mapsto a$ is called the *projection on* Σ and the monoid morphism from $(\Sigma \times D)^*$ to $\mathbb{N}^d$ defined by $(a, \overline{v}) \mapsto \overline{v}$ is called the *extended Parikh image*.

Remark. Let $\Sigma = \{a_1, \ldots, a_n\}$ and $D \subseteq \mathbb{N}^n$. If a word $\omega \in (\Sigma \times D)^*$ is in $\{(a_i, \overline{e_i}) \mid 1 \leq i \leq n\}^*$, then the extended Parikh image of ω is the Parikh image its projection on Σ.

Definition 1 (Parikh automaton [16]). Let Σ be an alphabet, $d \in \mathbb{N}^+$, and D a finite subset of $\mathbb{N}^d$. A *Parikh automaton (PA)* of dimension d over $\Sigma \times D$ is a pair (A, C) where A is a finite automaton over $\Sigma \times D$, and $C \subseteq \mathbb{N}^d$ is a semilinear set. The PA language, written $L(A, C)$, is the projection on Σ of the words of $L(A)$ whose extended Parikh image is in C. The PA is said to be *deterministic (DetPA)* if for every state q of A and every $a \in \Sigma$, there exists at most one pair $(q', \overline{v})$ with q' a state and $\overline{v} \in D$ such that $(q, (a, \overline{v}), q')$ is a transition of A. We write $\mathcal{L}_{\text{PA}}$ (resp. $\mathcal{L}_{\text{DetPA}}$) for the class of languages recognized by PA (resp. DetPA).

An alternative view of the PA will prove very useful. Indeed we note that a PA can be viewed equivalently as an automaton that applies a semilinear constraint on the counts of the individual transitions occurring along its accepting runs. To explain this, let (A, C) be a PA of dimension d, and let $\delta = \{t_1, \ldots, t_n\}$ be the transitions of A. Consider the automaton B which is a copy of A except that the vector part of the transitions is dropped, and suppose there is a natural bijection between the transitions of the two automata. Let π be a path in A; the contribution to the extended Parikh image of $\mu(\pi)$ of the transition $t_i = (p, (a, \overline{v_i}), q)$ is $\overline{v_i}$; thus, knowing how many times t_i appears in the path traced by π in B is enough to retrieve the value of the extended Parikh image of $\mu(\pi)$. Now note that the bijection exists if no two distinct transitions t_i, t_j are such that $t_i = (p, (a, \overline{v_i}), q)$ and $t_j = (p, (a, \overline{v_j}), q)$. However, if such t_i and t_j exist, we can replace them by $t = (p, (a, \overline{e_{d+1}}), q)$, incrementing in the process the dimension of PA, and change C to C' defined by $(\overline{v}, c) \in C' \Leftrightarrow (\exists c_i)(\exists c_j)[c = c_i + c_j \wedge \overline{v} + c_i.\overline{v_i} + c_j.\overline{v_j} \in C]$ without changing the language of the PA. It is thus readily seen that the following defines models equivalent to the PA[2] and the DetPA:

Definition 2 (Constrained automaton)**.** A *constrained automaton (CA)* over an alphabet Σ is a pair (A, C) where A is a finite automaton over Σ with d transitions, and $C \subseteq \mathbb{N}^d$ is a semilinear set. Its language is $L(A, C) = \{\mu(\pi) \mid \pi \in \mathsf{Run}(A) \wedge \Phi(\pi) \in C\}$. The CA is said to be *deterministic (DetCA)* if A is deterministic.

3.1. On the expressiveness of Parikh automata

The constrained automaton characterization of PA helps deriving pumping-style necessary conditions for membership in $\mathcal{L}_{\text{PA}}$ and in $\mathcal{L}_{\text{DetPA}}$:

Lemma 1. *Let $L \in \mathcal{L}_{\text{PA}}$. There exist $p, \ell \in \mathbb{N}^+$ such that any $w \in L$ with $|w| > \ell$ can be written as $w = uvxvz$ where:*
1. *$0 < |v| \le p$, $|x| > p$, and $|uvxv| \le \ell$,*
2. *$uv^2xz \in L$ and $uxv^2z \in L$.*

Proof. Let (A, C) be a CA of language L. Let p be the number of states in A and m be the number of elementary cycles (i.e., cycles in which no state except the start state occurs twice) in the underlying multigraph of A. Finally, let $\ell = p \times (2m + 1)$. Now, let $w \in L$ such that $|w| \ge \ell$ and $\pi \in \mathsf{Run}(A)$ such that $\mu(\pi) = w$ and $\Phi(\pi) \in C$. Write π as $\pi_1 \cdots \pi_{2m+1}\rho$ where $|\pi_i| = p$. By the pigeonhole principle, each π_i contains an elementary cycle, and thus, there exist $1 \le i, j \le m + 1$ with $i + 1 < j$ such that π_i and π_j share the same cycle η_v labeled with a word v. Write:
- π_i as $\pi_{i,1}\eta_v\pi_{i,2}$, and π_j as $\pi_{j,1}\eta_v\pi_{j,2}$,
- η_u for $\pi_1 \cdots \pi_{i-1}\pi_{i,1}$ and u for $\mu(\eta_u)$,
- η_x for $\pi_{i,2}\pi_{i+1} \cdots \pi_{j-1}\pi_{j,1}$ and x for $\mu(\eta_x)$,
- η_z for $\pi_{j,2}\pi_{j+1} \cdots \pi_{\ell+1}\rho$ and z for $\mu(\eta_z)$.

Then $\pi = \eta_u\eta_v\eta_x\eta_v\eta_z$ and $w = uvxvz$. Moreover, both $\pi' = \eta_u\eta_v^2\eta_x\eta_z$ and $\pi'' = \eta_u\eta_x\eta_v^2\eta_z$ are accepting paths with the same Parikh image as π. Thus, $\mu(\pi') = uv^2xz \in L$ and $\mu(\pi'') = uxv^2z \in L$. Moreover, $0 < |v| \le p, |x| > p$ and $|uvxv| \le \ell$. □

A similar argument leads to a stronger property for the languages belonging to $\mathcal{L}_{\text{DetPA}}$:

Lemma 2. *Let $L \in \mathcal{L}_{\text{DetPA}}$. There exist $p, \ell \in \mathbb{N}^+$ such that any w over the alphabet of L with $|w| > \ell$ can be written as $w = uvxvz$ where:*
1. *$0 < |v| \le p$, $|x| > p$ and $|uvxv| \le \ell$,*
2. *$uv^2x, uvxv$ and uxv^2 are equivalent w.r.t. the Nerode relation of L.*

[2] Another equivalent view of PA languages suggested by one referee is as sets $R^{-1}(X)$ where R is a rational relation over $\Sigma^* \times \mathbb{N}^d$ and X is a rational subset of $\mathbb{N}^d$. An artificial further restriction to this viewpoint would serve to capture DetPA languages.

We apply Lemma 1 to the language COPY, defined as $\{w\#w \mid w \in \{a,b\}^*\}$, as follows:

Proposition 3. COPY $\notin \mathcal{L}_{\text{PA}}$.

Proof. Suppose COPY $\in \mathcal{L}_{\text{PA}}$. Let ℓ, p be given by Lemma 1, and consider $w = (a^p b)^\ell \# (a^p b)^\ell \in$ COPY. Lemma 1 states that $w = uvxvz$ where $uvxv$ lays in the first half of w, and $s = uv^2xz \in$ COPY. Note that x contains at least one b. Suppose $v = a^i$ for $1 \le i \le p$, then there is a sequence of a's in the first half of s unmatched in the second half. Likewise, if v contains a b, then s has a sequence of a's between two b's unmatched in the second half. Thus $s \notin$ COPY, a contradiction. Hence COPY $\notin \mathcal{L}_{\text{PA}}$. □

As Klaedtke and Rueß show using closure properties, DetPA are strictly weaker than PA. The thinner grain of Lemma 2 suggests explicit languages that witness the separation of $\mathcal{L}_{\text{DetPA}}$ from $\mathcal{L}_{\text{PA}}$. Indeed, let EQUAL $\subseteq \{a,b,\#\}^*$ be the language $\{a,b\}^* \cdot \{a^n\#a^n \mid n \in \mathbb{N}\}$, we have:

Proposition 4. EQUAL $\in \mathcal{L}_{\text{PA}} \setminus \mathcal{L}_{\text{DetPA}}$.

Proof. We omit the proof that EQUAL $\in \mathcal{L}_{\text{PA}}$. Now, suppose EQUAL $\in \mathcal{L}_{\text{DetPA}}$, and let ℓ, p be given by Lemma 2. Consider $w = (a^p b)^\ell$. Lemma 2 then asserts that a prefix of w can be written as $w_1 = uvxv$, and that $w_2 = uv^2x$ verifies $w_1 \equiv w_2$. As $|x| > p$, x contains a b. Let k be the number of a's at the end of w_1. Suppose $v = a^i$ for $1 \le i \le p$, then w_2 ends with $k - i < k$ letters a. Thus $w_1\#a^k \in$ EQUAL and $w_2\#a^k \notin$ EQUAL, a contradiction. Suppose then that $v = a^i b a^k$, with $0 \le i + k < p$. Then w_2 ends with $p - i > k$ letters a, and similarly, $w_1 \not\equiv w_2$, a contradiction. Thus EQUAL $\notin \mathcal{L}_{\text{DetPA}}$. □

For comparison, we mention another line of attack for the study of $\mathcal{L}_{\text{DetPA}}$. The proof is omitted, but is based on the number of possible configurations of a PA, which is polynomial in the length of the input word. Klaedtke and Rueß used a similar argument to show that PAL $= \{w\#w^R \mid w \in \{a,b\}^+\}$, where w^R is the reversal of w, is not in $\mathcal{L}_{\text{PA}}$.

Lemma 5. *Let $L \in \mathcal{L}_{\text{DetPA}}$. Then there exists $c > 0$ such that $|\{[w]_L \mid w \in \Sigma^n\}| \in O(n^c)$.*

Proposition 6. *Let $L = \{w \in \{a,b\}^* \mid w_{|w|_a} = b\}$, where w_i is the i-th letter of w. Then $L \in \mathcal{L}_{\text{PA}} \setminus \mathcal{L}_{\text{DetPA}}$.*

Proof. We omit the proof that $L \in \mathcal{L}_{\text{PA}}$; the main point is simply to guess the position of the b referenced by $|w|_a$. On the other hand, let $n > 0$ and $u, v \in \{a,b\}^n$ such that $|u|_a = |v|_a = \frac{n}{2}$ and there exists $p \in \{\frac{n}{2}, \ldots, n\}$ with $u_p \neq v_p$. Let $w = a^{p-\frac{n}{2}}$, then $(uw)_{|uw|_a} = (uw)_{|u|_a + |w|_a} = (uw)_p = u_p$, and similarly, $(vw)_{|vw|_a} = v_p$. This implies $uw \notin L \leftrightarrow vw \in L$, thus $u \not\equiv v$. Then for $0 \le i \le \frac{n}{2}$, define $E_i = \{a^{\frac{n}{2}-i} b^i z \mid z \in \{a,b\}^{\frac{n}{2}} \wedge |z|_a = i\}$. For any $u, v \in \bigcup E_i$ with $u \neq v$, the previous discussion shows that $u \not\equiv v$. Thus $|\{[w]_L \mid w \in \{a,b\}^n\}| \ge |\bigcup_{i=0}^{\frac{n}{2}} E_i| = \sum_{i=0}^{\frac{n}{2}} |E_i| = \sum_{i=0}^{\frac{n}{2}} \binom{\frac{n}{2}}{i} = 2^{\frac{n}{2}} \notin O(n^{O(1)})$. Lemma 5 then implies that $L \notin \mathcal{L}_{\text{DetPA}}$. □

3.2. On decidability and closure properties of Parikh automata

The following table summarizes decidability results for PA and DetPA. The results in bold are new, while the others are from [16] and [12]:

	$=\emptyset$	$=\Sigma^*$	is finite	$\subseteq$	is regular
DetPA	D	D	**D**	**D**	?
PA	D	U	**D**	**U**	**U**

Proposition 7. *(1) Finiteness is decidable for PA. (2) Inclusion is decidable for DetPA and undecidable for PA. (3) Regularity is undecidable for PA.*

Proof. (1). Let (A, C) be a CA. Then $\mathsf{Run}(A)$ is a regular language, and thus, its Parikh image is effectively semilinear (this is a special case of Parikh's theorem [20]). It follows that the language described by A and C is finite if and only if $\Phi(\mathsf{Run}(A)) \cap C$ is finite, which is decidable. *(2).* Decidability of inclusion for DetPA follows from the fact that $\mathcal{L}_{\text{DetPA}}$ is closed under complement and intersection, and that the emptiness problem is decidable for DetPA. (In fact, it is decidable whether the language of a PA is included in the language of a DetPA.) Undecidability of inclusion for PA follows immediately from the undecidability of the universe problem for PA. *(3).* This follows from a theorem of [11], which states the following: Let $\mathcal{C}$ be a class of languages closed under union and under concatenation with regular languages. Let P be a predicate on languages true of every regular language, false of some languages, preserved by inverse rational transduction, union with $\{\varepsilon\}$ and intersection with regular languages. Then P is undecidable in $\mathcal{C}$. Obviously, $\mathcal{L}_{\text{PA}}$ satisfies the hypothesis for $\mathcal{C}$. Moreover, "being regular in $\mathcal{L}_{\text{PA}}$" is a predicate satisfying the hypothesis for P. Thus, regularity is undecidable for PA. ☐

We now further the study of closure properties of PA and DetPA started in [16]. The following table collects the closure properties of PA and DetPA, where h is a morphism, c is the commutative closure. In bold are the results of the present paper, while the other results can be found in [16] (detailed proofs by Karianto can be found in [14]):

	$\cup$	$\cap$	$\cdot$	$^{-}$	h	h^{-1}	c	*
DetPA	Y	Y	**N**	Y	**N**	Y	**Y**	**N**
PA	Y	Y	Y	N	Y	Y	**Y**	N

As the language EQUAL separating $\mathcal{L}_{\text{DetPA}}$ from $\mathcal{L}_{\text{PA}}$ is the concatenation of a regular language and a language of $\mathcal{L}_{\text{DetPA}}$, we have:

Proposition 8. $\mathcal{L}_{\text{DetPA}}$ *is not closed under concatenation.*

Proposition 9. *(1) The commutative closure of any semilinear language is in* $\mathcal{L}_{\text{DetPA}}$. *(2)* $\mathcal{L}_{\text{DetPA}}$ *is not closed under morphisms.*

Proof. (1). Let $\Sigma = \{a_1, \ldots, a_n\}$, $L \subseteq \Sigma^*$ a semilinear language, and $C = \Phi(L)$. Define A to be an automaton with one state, initial and final, with n loops, the i-th labeled $(a_i, \overline{e_i}) \in \Sigma \times \{\overline{e_i}\}_{1 \leq i \leq n}$. Then $c(L) = L(A, C)$. *(2)* is straightforward as any language of $\mathcal{L}_{\text{PA}}$ is the image by a morphism of a language in $\mathcal{L}_{\text{DetPA}}$. Indeed, say (A, C) is a CA and let B be the copy of A in which the transition t is relabeled t; then B is deterministic and $L(A, C) = \mu_A(L(B, C))$. This implies the nonclosure of $\mathcal{L}_{\text{DetPA}}$ under morphisms. ☐

Note that *(1)* from Proposition 9 implies that both $\mathcal{L}_{\text{PA}}$ and $\mathcal{L}_{\text{DetPA}}$ are closed under commutative closure, as both are classes of semilinear languages [16].

Proposition 10. *Neither $\mathcal{L}_{\text{PA}}$ nor $\mathcal{L}_{\text{DetPA}}$ is closed under starring.*

Proof. We show that the starring of $L = \{a^n b^n \mid n \in \mathbb{N}\}$ is not in $\mathcal{L}_{\text{PA}}$. Suppose $L^* \in \mathcal{L}_{\text{PA}}$, and let $w = (a^p b^p)^\ell$, where ℓ, p are given by Lemma 1. The same lemma asserts that $w = uvxvz$, such that, in particular, uv^2xz and uxv^2z are in L^*. Now suppose $v = a^i$ for some $i \leq p$. Then uv^2x contains $a^{p+i}b^p$ with no more b's on the right. Thus $uv^2xz \notin L^*$. The case for $v = b^i$ is similar. Now suppose $v = a^i b^j$ with $i, j > 0$. Then uv^2x contains $\cdots a^p b^j a^i b^p \cdots$, but $i < p$, thus $uv^2xz \notin L^*$. The case $v = b^i a^j$ is similar. Thus $L^* \notin \mathcal{L}_{\text{PA}}$. □

Remark. Baker and Book [1] already note, in different terms, that if $\mathcal{L}_{\text{PA}}$ were closed under starring, it would be an intersection closed full AFL containing $\{a^n b^n \mid n \geq 0\}$, and so would be equal to the class of Turing-recognizable languages. Thus $\mathcal{L}_{\text{PA}}$ is not closed under starring.

3.3. Parikh automata and reversal-bounded counter machines

Klaedtke and Rueß noticed in [15] that Parikh automata recognize the same languages as reversal-bounded counter machines, a model introduced by Ibarra [12]:

Definition 3 (Reversal-bounded counter machine [12])**.** A *one-way, k-counter machine M* is a 5-uple $(Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, Σ is an alphabet, $\delta \subseteq Q \times (\Sigma \cup \{\sharp\}) \times \{0,1\}^k \times Q \times \{S, R\} \times \{-1, 0, +1\}^k$ is the transition function, $q_0 \in Q$ is the initial state and $F \subseteq Q$ is the set of final states. Moreover, we suppose $\sharp \notin \Sigma$. The machine is *deterministic* if for any $(p, \ell, \overline{x})$, there exists at most one $(q, h, \overline{v})$ such that $(p, \ell, \overline{x}, q, h, \overline{v}) \in \delta$. On input w, the machine starts with a read-only tape containing $w\sharp$, and its head on the first character of w. Write c_i for the i-th counter, then a transition $(p, \ell, \overline{x}, q, h, \overline{v}) \in \delta$ is taken if the machine is in state p, reading character ℓ and $c_i = 0$ if $x_i = 0$ and $c_i > 0$ if $x_i = 1$, for all i. The machine then enters state q, its head is moved to the right iff $h = R$, and $\overline{v}$ is added to the counters. If the head falls off the tape, or if a counter turns negative, the machine rejects. A word is accepted if an execution leads to a final state. The machine is *reversal-bounded (RBCM)* if there exists an integer r such that any accepting run changes between increments and decrements of the counters a (bounded) number of times less than r. We write *DetRBCM* for deterministic RBCM. We write $\mathcal{L}_{\text{RBCM}}$ (resp. $\mathcal{L}_{\text{DetRBCM}}$) for the class of languages recognized by RBCM (resp. DetRBCM).

In [15, Section A.3], it is shown that PA have the same expressive power as (nondeterministic) RBCM. Although Fact 30 of [15], on which the authors rely to prove this result, is technically false as stated,[3] the small gap there can be fixed so that:

Proposition 11 ([15])**.** $\mathcal{L}_{\text{PA}} = \mathcal{L}_{\text{RBCM}}$.

[3] Fact 30 of [15] states the following. Consider a RBCM M which, for any counter, changes between increment and decrement only once. Let M' be M in which negative counter values are allowed and the zero-tests are ignored. Then a word is claimed to be accepted by M iff the run of M' on the same word reaches a final state with all its counters nonnegative. A counter-example is the following. Take A to be the minimal automaton for a^*b, and add a counter for the number of a's that blocks the transition labeled b unless the counter is nonzero. This machine recognizes a^+b. Then by removing this test, the machine now accepts b.

Further, we study how the notion of determinism compares in the two models. Let NSUM $= \{a^n\spadesuit b^{m_1}\#b^{m_2}\#\cdots\#b^{m_k}\clubsuit c^{m_1+\cdots+m_n} \mid k \geq n \geq 0 \wedge (\forall i)[m_i \in \mathbb{N}]\}$: the number of a's is the number of m_i's to add to get the number of c's. Note that NSUM is not context-free. Then:

Proposition 12. *$\mathcal{L}_{\text{DetPA}} \subsetneq \mathcal{L}_{\text{DetRBCM}}$ and* NSUM $\in \mathcal{L}_{\text{DetRBCM}} \setminus \mathcal{L}_{\text{DetPA}}$.

Proof. We first show that $\mathcal{L}_{\text{DetPA}} \subseteq \mathcal{L}_{\text{DetRBCM}}$. Let (A, C) be a CA, where $A = (Q, \Sigma, \delta, q_0, F)$ is deterministic and let $\delta = \{t_1, \ldots, t_k\}$. We define a DetRBCM of the same language in two steps. (1) First, let M be the k-counter machine $(Q \cup \{q_f\}, \Sigma, \zeta, q_0, q_f)$, where $q_f \notin Q$ and ζ is defined by:

$$\zeta = \bigcup_{\overline{x}\in\{0,1\}^k} \Big(\big\{(q, a, \overline{x}, q', R, \overline{e_i}) \mid t_i = (q, a, q')\big\} \cup \big\{(q, \sharp, \overline{x}, q_f, S, \overline{0}) \mid q \in F\big\} \Big).$$

This machine (trivially a DetRBCM) does not make any test, and accepts (in q_f) precisely the words accepted by A. Moreover, the state of the counters in q_f is the Parikh image of the path taken (in A) to recognize the input word. (2) We then refine M to check that the counter values belong to C. We note that we can do that as a direct consequence of the proof of [13, Theorem 3.5], but this proof relied on nontrivial algebraic properties of systems $A\overline{y} = \overline{b}$, where A is a matrix, $\overline{y}$ are unknowns and $\overline{b}$ is a vector; we present here an elementary proof. Recall that C can be expressed as a quantifier-free first-order formula which uses the function symbol $+$, the congruence relations $\equiv_i$, for $i \geq 2$, and the order relation $<$ (see, e.g., [7]). So let C be given as such formula ϕ_C with k free variables. Let ϕ_C be put in disjunctive normal form. The machine M then tries each and every clause of ϕ_C for acceptance. First, note that a term can be computed with a number of counters and reversals which depends only on its size: for instance, computing $c_i + c_j$ requires two new counters x, y; c_i is decremented until it reaches 0, while x and y are incremented, so that their value is c_i; now decrement y until it reaches 0 while incrementing c_i back to its original value; then do the same process with c_j: as a result, x is now $c_i + c_j$. Second, note that any atomic formula ($t_1 < t_2$ or $t_1 \equiv_i t_2$) can be checked by a DetRBCM: for $t_1 < t_2$, compute $x_1 = t_1$ and $x_2 = t_2$, then decrement x_1 and x_2 until one of them reaches 0, if the first one is x_1, then the atomic formula is true, and false otherwise; for $t_1 \equiv_i t_2$, a simple automaton-based construction depending on i can decide if the atomic formula is true. Thus, a DetRBCM can decide, for each clause, if all of its atomic formulas (or negation) are true, and in this case, accept the word. This process does not use the read-only head, and uses a number of counters and a number of reversals bounded by the length of ϕ_C.

We now show that NSUM $\in \mathcal{L}_{\text{DetRBCM}} \setminus \mathcal{L}_{\text{DetPA}}$. We omit the fact that NSUM $\in \mathcal{L}_{\text{DetRBCM}}$. Now suppose (A, C) is a DetPA such that $L(A, C) =$ NSUM, with $A = (Q, \Sigma \times D, \delta, q_0, F)$ also deterministic. We may suppose that the projection on Σ of $L(A)$ is a subset of $a^*\spadesuit(b^*\#)^*b^*\clubsuit c^*$, so that there exist $k \geq 0$, $q_1, \ldots, q_k \in Q$, and $j \in \{0, \ldots, k\}$ such that $(q_i, (a, \overline{v_i}), q_{i+1}) \in \delta$, for $0 \leq i < k$ and some $\overline{v_i}$'s, and $(q_k, (a, \overline{v_k}), q_j) \in \delta$. Moreover, we may suppose that no other transition points to one of the q_i's, and that all transitions $t = (q_i, (\ell, \overline{v}), q) \in \delta$ such that $q \notin \{q_0, \ldots, q_k\}$ are with $\ell = \spadesuit$; let T be the set of all such transitions t. We define $|T|$ DetPA such that the union of their languages is SUMN $= \{\spadesuit w \heartsuit a^n \mid a^n\spadesuit w \in \text{NSUM}\}$, that is, the strings of NSUM with a^n pushed at the end. For $t \in T$, define A_t as the automaton similar to A but which starts with the transition t and delay the first part of the computation until the very end. Formally, $A_t = (Q \cup \{q'_0\}, \Sigma \times D, \delta_t, q'_0, \{\mathsf{From}(t)\})$ where $\delta_t = (\delta \setminus T) \cup \{(q'_0, \mu(t), \mathsf{To}(t)\} \cup \{(q_f, (\heartsuit, \overline{0}), q_0) \mid q_f \in F\}$ with q'_0 a fresh state. Now for $\omega \in L(A)$, let t be the transition labeled ♠ taken when A reads ω, and let $\omega = \omega_1\mu(t)\omega_2$. Then $\mu(t)\omega_2(\heartsuit, \overline{0})\omega_1 \in L(A_t)$, and this word has the same extended Parikh image as ω. Thus we have that $\bigcup_{t \in T} L(A_t, C) = \text{SUMN}$, and if NSUM $\in \mathcal{L}_{\text{DetPA}}$, then SUMN $\in \mathcal{L}_{\text{DetPA}}$. A proof similar to Proposition 4 then shows that SUMN $\notin \mathcal{L}_{\text{DetPA}}$, a contradiction; thus NSUM $\notin \mathcal{L}_{\text{DetPA}}$. □

The parallel drawn between (Det)PA and (Det)RBCM allows transferring some RBCM and DetRBCM results to PA and DetPA. An example is a consequence of the following lemma proved in 2011 by Chiniforooshan *et al.* [5] for the purpose of showing incomparability results between different models of reversal-bounded counter machines:

Lemma 13 ([5])**.** *Let a DetRBCM express $L \subseteq \Sigma^*$. Then there exists $w \in \Sigma^*$ such that $L \cap w\Sigma^*$ is a nontrivial regular language.*

Variants of the language EQUAL from Proposition 4 can be shown outside $\mathcal{L}_{\text{DetPA}}$ in this way. For instance, for $\Sigma = \{a, b\}$, $\Sigma\text{ANBN} = \Sigma^* \cdot \{a^n b^n \mid n \in \mathbb{N}\}$ is such that any $w \in \Sigma^*$ makes $\Sigma\text{ANBN} \cap w\Sigma^*$ nonregular. Although Lemma 13 thus gives languages in $\mathcal{L}_{\text{PA}} \setminus \mathcal{L}_{\text{DetPA}}$, Lemma 13 seemingly does not apply to EQUAL itself since $\text{EQUAL} \cap \#\{a, b, \#\}^* = \{\#\}$ is regular.

4. Affine Parikh automata

A PA of dimension d can be viewed as an automaton in which each transition updates a vector $\overline{x}$ of $\mathbb{N}^d$ using a function $\overline{x} \leftarrow \overline{x} + \overline{v}$ where $\overline{v}$ depends only on the transition. At the end of an accepting computation, the word is accepted if $\overline{x}$ belongs to some semilinear set. We propose to generalize the updating function to an affine function. We start by defining the model, and show that defining it over $\mathbb{N}$ is at least as general as defining it on $\mathbb{Q}$. We study the expressiveness of this model, and show it is strictly more powerful than PA. We then note that deterministic such automata can be normalized so as to essentially trivialize their automaton component. We then study nonclosure properties and decidability problems associated with APA, leading to the observation that APA lack some desirable properties — e.g., properties usually needed for any real-world application.

In the following, we consider the vectors in $\mathbb{K}^d$ to be *column* vectors. Let $d, d' > 0$. A function $f : \mathbb{K}^d \to \mathbb{K}^{d'}$ is a (total) *affine function* if there exist a matrix $M \in \mathbb{K}^{d' \times d}$ and $\overline{v} \in \mathbb{K}^{d'}$ such that for any $\overline{x} \in \mathbb{K}^d$, $f(\overline{x}) = M.\overline{x} + \overline{v}$; it is *linear* if $\overline{v} = \overline{0}$. We note such a function $f = (M, \overline{v})$. We write $\mathcal{F}_d^{\mathbb{K}}$ for the set of affine functions from $\mathbb{K}^d$ to $\mathbb{K}^d$ and view $\mathcal{F}_d^{\mathbb{K}}$ as the monoid $(\mathcal{F}_d^{\mathbb{K}}, \diamond)$ with $(f \diamond g)(\overline{x}) = g(f(\overline{x}))$.

Definition 4 (Affine Parikh automaton)**.** A $\mathbb{K}$-*affine Parikh automaton ($\mathbb{K}$-APA)* of dimension d is a triple (A, U, C) where A is an automaton with transition set δ, U is a morphism from δ^* to $\mathcal{F}_d^{\mathbb{K}}$ and $C \subseteq \mathbb{K}^d$ is a $\mathbb{K}$-definable set; recall that U need only be defined on δ. The language of the APA is $L(A, U, C) = \{\mu(\pi) \mid \pi \in \mathsf{Run}(A) \wedge (U(\pi))(\overline{0}) \in C\}$. The $\mathbb{K}$-APA is said to be *deterministic ($\mathbb{K}$-DetAPA)* if A is. We write $\mathcal{L}_{\mathbb{K}\text{-APA}}$ (resp. $\mathcal{L}_{\mathbb{K}\text{-DetAPA}}$) for the class of languages recognized by $\mathbb{K}$-APA (resp. $\mathbb{K}$-DetAPA).

Remark. It is easily seen that $\mathbb{N}$-APA (resp. $\mathbb{N}$-DetAPA) are a generalization of CA (resp. DetCA). Indeed, let (A, C) be a CA, and let Φ be the Parikh image over the set δ of transitions of A. Define, for $t \in \delta$, $U(t) = (Id, \Phi(t))$ where Id is the identity matrix of dimension $|\delta| \times |\delta|$. Then $L(A, C) = L(A, U, C)$; we will later see that this containment is strict.

The arguments used by Klaedtke and Rueß [15] apply equally well to $\mathbb{K}$-APA and $\mathbb{K}$-DetAPA, showing:

Proposition 14. *$\mathcal{L}_{\mathbb{K}\text{-APA}}$ and $\mathcal{L}_{\mathbb{K}\text{-DetAPA}}$ are effectively closed under union, intersection and inverse morphisms. Moreover, $\mathcal{L}_{\mathbb{K}\text{-APA}}$ is closed under concatenation and nonerasing morphisms, and $\mathcal{L}_{\mathbb{K}\text{-DetAPA}}$ is closed under complement.*

We now show these models over $\mathbb{N}$ are at least as powerful as over $\mathbb{Q}$. First, we need the following technical lemma:

Lemma 15. *For any $\mathbb{K}$-APA (resp. $\mathbb{K}$-DetAPA) there exists a $\mathbb{K}$-APA (resp. $\mathbb{K}$-DetAPA) where the functions associated with the transitions are linear, except for some transitions which can be taken only as the first transition of a nonempty run.*

Proof (sketch). Let (A, U, C) be a $\mathbb{K}$-APA of dimension d, where the transition set of A is $\delta = \{t_1, \ldots, t_{|\delta|}\}$, and write $U(t_i) = (M_i, \overline{v_i})$. Let A' be a copy of A in which a fresh state q is added, set to be the initial state, with the same outgoing transitions as the initial state of A and no incoming transition. Let $t'_1, \ldots, t'_k$ be the new transitions in A', and order δ such that $t_1, \ldots, t_k$ are the corresponding transitions leaving the initial state of A. Now define U', for $\overline{x}, \overline{y_1}, \ldots, \overline{y_{|\delta|}} \in \mathbb{K}^d$, by $U'(t'_i)\colon (\overline{x}, \overline{y_1}, \ldots, \overline{y_{|\delta|}}) \mapsto (\overline{v_i}, \overline{v_1}, \ldots, \overline{v_{|\delta|}})$, and $(U'(t_i)\colon (\overline{x}, \overline{y_1}, \ldots, \overline{y_{|\delta|}}) \mapsto (M_i.\overline{x} + \overline{y_i}, \overline{y_1}, \ldots, \overline{y_{|\delta|}})$. Finally define $C' = C.\mathbb{K}^{d \times |\delta|}$. Then $L(A', U', C') = L(A, U, C)$, and A' is deterministic if A is. Moreover, the only nonlinear functions given by U' are for the outgoing transitions of the initial state of A', a state no run can return to. $\square$

Proposition 16. $\mathcal{L}_{\mathbb{Q}\text{-DetAPA}} \subseteq \mathcal{L}_{\mathbb{N}\text{-DetAPA}}$ *and* $\mathcal{L}_{\mathbb{Q}\text{-APA}} \subseteq \mathcal{L}_{\mathbb{N}\text{-APA}}$.

Proof. We first recall that a set $C \subseteq \mathbb{Q}^d$ is $\mathbb{Q}$-definable iff it is a finite union of sets of the form:

$$\{\overline{x} \mid f_1(\overline{x}) = \cdots = f_p(\overline{x}) = 0 \wedge g_1(\overline{x}) > 0 \wedge \cdots \wedge g_q(\overline{x}) > 0\},$$

where $f_1, \ldots, f_p, g_1, \ldots, g_q\colon \mathbb{Q}^d \to \mathbb{Q}$ are affine functions (see, e.g., [6]). Let (A, U, C) be a $\mathbb{Q}$-APA of dimension d; by Lemma 15, we may suppose that the functions associated with the transitions are linear, except for the transitions that may begin a run. We suppose C is a single set of the kind previously described; this is no loss of generality as $\mathcal{L}_{\mathbb{K}\text{-APA}}$ and $\mathcal{L}_{\mathbb{K}\text{-DetAPA}}$ are closed under union. So let C be described by functions f_i and g_i as above, and suppose $d = p + q$ (we add constant 0 functions to the f_i's or 0's to the vectors of C in order to do that). Define $f : \mathbb{Q}^d \to \mathbb{Q}^d$ by $f(\overline{x}) = (f_1(\overline{x}), \ldots, g_1(\overline{x}), \ldots)$; clearly, $f \in \mathcal{F}_d(\mathbb{Q})$. Now let (A, U', C') be the $\mathbb{Q}$-APA of dimension $2d$, defined by $(U'(t))(\overline{x}, \overline{y}) = ((U(t))(\overline{x}), f(\overline{x}))$, with t a transition of A and $\overline{x}, \overline{y} \in \mathbb{Q}^d$; and $C' = \mathbb{Q}^d.\{0\}^p.(\mathbb{Q}^+)^q$. Clearly, $L(A, U', C') = L(A, U, C)$. We then define U'' by $U''(t) = c \times U'(t)$ where c is the maximum denominator in the reduced fractions appearing in the matrix and vector of $U'(t)$. Thus, the functions given by U'' are from $\mathbb{Z}^{2d}$ to $\mathbb{Z}^{2d}$. Moreover, for any $\pi \in \mathsf{Run}(A)$, $(U''(\pi))(\overline{0}) = k \times (U'(\pi))(\overline{0})$, for some $k \in \mathbb{N}^+$

depending only on π. Thus, defining $C'' = \mathbb{Z}^d.\{0\}^p.(\mathbb{Z}^+)^q$, we have $L(A, U'', C'') = L(A, U', C')$. Finally, the negative numbers can be circumvented by doubling the dimension of the matrices and keeping track of the negative and the positive contributions separately until the final tests for zero, which become tests that negative contribution equals (or is strictly lesser than) the positive contribution of a number (a similar technique is used by Klaedtke and Rueß [15]). □

Remark. The previous proof shows that the constraint set of $\mathbb{Q}$-APA can be simulated *within* the automaton, and is thus of a lesser use.

We now give a large class of languages belonging to $\mathcal{L}_{\mathbb{Q}\text{-APA}}$. Define $\mathcal{M}_\cap(L)$ as the smallest semiAFL containing L and closed under intersection; that is, $\mathcal{M}_\cap(L)$ is the smallest class of languages containing L and closed under nonerasing and inverse morphism, intersection with a regular set, union, intersection, and concatenation. With $\text{PAL} = \{w\#w^R \mid w \in \{a,b\}^+\}$:

Proposition 17. $\mathcal{M}_\cap(\text{PAL}) \subseteq \mathcal{L}_{\mathbb{Q}\text{-APA}}$.

Proof. We sketch a $\mathbb{Q}$-DetAPA for PAL. The automaton starts by reading a single letter, if it is an a it initializes its counters to $(2,1)$, otherwise, it initializes them to $(2,0)$. Now for each letter read, if it is an a, it applies the function $(p,v) \mapsto (2p, v+p)$, and $(p,v) \mapsto (2p, v)$ if it is a b. Upon reaching the # sign, functions associated to a and b change: when reading an a, the automaton applies $(p,v) \mapsto (p/2, v-p/2)$, otherwise it applies $(p,v) \mapsto (p/2, v)$. Clearly, a word is in PAL iff it is of the form $\{a,b\}^+\#\{a,b\}^+$ and the final state of the counters is $(1,0)$. The closure properties are implied by those of $\mathcal{L}_{\mathbb{Q}\text{-APA}}$ (Proposition 14). □

The class $\mathcal{M}_\cap(\text{PAL})$ contains a wide range of languages. First, the closure of PAL under nonerasing and inverse morphism and intersection with regular sets is the class of *linear languages* (e.g., [4]). In turn, adding closure under intersection permits to express the languages of nondeterministic multipushdown automata where in every computation, each pushdown store makes a bounded number of reversals (that is, going from pushing to popping) [3]; in particular, if there is only one such pushdown store, this corresponds to the *ultralinear languages* [10]. Further, as $\mathcal{M}_\cap(\text{COPY}) \subsetneq \mathcal{M}_\cap(\text{PAL})$ (e.g., [4]) this implies that $\text{COPY} \in \mathcal{L}_{\mathbb{Q}\text{-APA}}$.

Next, we note that $\mathbb{K}$-APA express only context-sensitive languages (CSL):

Proposition 18. $\mathcal{L}_{\mathbb{N}\text{-APA}} \subseteq \text{CSL}$.

Proof. Let (A, U, C) be an $\mathbb{N}$-APA of dimension d, we show that $L(A,U,C) \in \text{NSPACE}[n]$ (which is equal to CSL [18]). Let $A = (Q, \Sigma, \delta, q_0, F)$, and $w = w_1 \cdots w_n \in \Sigma^*$. First, initialize $\overline{v} \leftarrow \overline{0}$ and $q \leftarrow q_0$. Iterate through the letters of w: on the i-th letter, choose nondeterministically a transition t from q labeled with w_i. Update $\overline{v}$ by setting $\overline{v} \leftarrow (U(t))(\overline{v})$ and q with $q \leftarrow \mathsf{To}(t)$. Upon reaching the last letter of w, accept w iff $q \in F$ and $\overline{v} \in C$.

We now bound the value of $\overline{v}$. Let c be the greatest value appearing in any of the matrices or vectors in $U(t)$, for any t. For a given $\overline{v}$, let $\max \overline{v}$ be $\max\{v_1, \ldots, v_d\}$. Then for any t, $((U(t))(\overline{v}))_i \leq d \times (c \times \max \overline{v}) + c$. Let π be a path, we then have that $((U(\pi))(\overline{0}))_i \leq (c(d+1))^{n-1}c$, thus the size of $\overline{v}$ at the end of the algorithm is in $O(n)$. Now note that, as C is semilinear, the language of the binary encoding of its elements is regular [22], and thus, checking $\overline{v} \in C$ can be done efficiently. Hence the given algorithm is indeed in $\text{NSPACE}[n]$. □

We now note that the power of $\mathbb{K}$-DetAPA does not owe to their capabilities as automata:

Proposition 19. *Let Σ be an alphabet. There exists a two-state automaton A_Σ such that for any $\mathbb{K}$-DetAPA over Σ, there exists a $\mathbb{K}$-DetAPA accepting the same language whose underlying automaton is A_Σ.*

Proof. Let (A, U, C) be a $\mathbb{K}$-DetAPA of dimension d where $A = (Q, \Sigma, \delta, q_0, F)$, with $Q = \{1, \ldots, k\}$ and $\Sigma = \{a_1, \ldots, a_m\}$. Let $N = k(d+1)$, we show that there exist $f_{a_1}, \ldots, f_{a_m} \in \mathcal{F}_N^{\mathbb{K}}$, a $\mathbb{K}$-definable set $G \subseteq \mathbb{K}^N$ and $\overline{o} \in \mathbb{K}^N$ such that:

$$w = \ell_1 \cdots \ell_{|w|} \in L(A, U, C) \quad \Leftrightarrow \quad f_{\ell_{|w|}} \circ \cdots \circ f_{\ell_1}(\overline{o}) \in G. \tag{1}$$

Our goal is to represent the state in which the $\mathbb{K}$-DetAPA is with a vector of size N. This vector is composed of k smaller vectors of size $(d+1)$. On taking a path π in A, let $q = \mathsf{To}(\pi)$ and $\overline{v} = (U(\pi))(0^d)$; then q and $\overline{v}$ describe the current configuration of the $\mathbb{K}$-DetAPA. Thus we define, for any $q \in Q$ and $\overline{v} \in \mathbb{K}^d$: $\mathsf{Vec}(q, \overline{v}) = (0^{d+1} \ \cdots \ 0^{d+1} \ \underbrace{1 \ \overline{v}}_{q\text{-th subvector}} \ 0^{d+1} \ \cdots \ 0^{d+1})$.

Now, for $t \in \delta$, let M_t and $\overline{b}_t$ be such that $U(t) = (M_t, \overline{b}_t)$. For the purpose of describing the matrix U_a below, when $t \notin \delta$ we let M_t stand for the all-zero matrix of dimension $d \times d$ and $\overline{b}_t$ be the all-zero vector of dimension d. Let χ be the characteristic function of δ. For $a \in \Sigma$, define:

$$U_a = \left(\begin{array}{c|c|c|c|c}
\chi((1,a,1)) & 0 \cdots 0 & \cdots & \chi((k,a,1)) & 0 \cdots 0 \\ \hline
\overline{b}_{(1,a,1)} & M_{(1,a,1)} & \cdots & \overline{b}_{(k,a,1)} & M_{(k,a,1)} \\ \hline
\vdots & \vdots & \ddots & \vdots & \vdots \\ \hline
\chi((1,a,k)) & 0 \cdots 0 & \cdots & \chi((k,a,k)) & 0 \cdots 0 \\ \hline
\overline{b}_{(1,a,k)} & M_{(1,a,k)} & \cdots & \overline{b}_{(k,a,k)} & M_{(k,a,k)}
\end{array}\right)$$

The matrix U_a is such that for $(p, a, q) \in \delta$ and $\overline{v} \in \mathbb{K}^d$, $U_a.\mathsf{Vec}(p, \overline{v}) = \mathsf{Vec}(q, M_{(p,a,q)}.\overline{v} + \overline{b}_{(p,a,q)})$. In other words, U_a computes the transition function and, according to the current state, applies the right affine function. More generally, for a path π in A starting at q_0 and labeled by $w = \ell_1 \cdots \ell_{|w|}$, we have $U_{\ell_{|w|}} \cdots U_{\ell_1}.\mathsf{Vec}(q_0, 0^d) = \mathsf{Vec}(\mathsf{To}(\pi), (U(\pi))(0^d))$, where 0^d is the all-zero vector of dimension d. We then let G be the $\mathbb{K}$-definable set which contains $\mathsf{Vec}(q, \overline{v})$ iff q is final and $\overline{v} \in C$: $G = \bigcup_{i \in F} \bigcup_{\overline{v} \in C} \mathsf{Vec}(i, \overline{v})$.

Now let $f_{a_i} \in \mathcal{F}_N^{\mathbb{K}}$ be defined as $(U_{a_i}, 0^N)$ and let $\overline{o} = \mathsf{Vec}(q_0, 0^d)$. Then we have precisely Equation (1). Now let A' be the automaton $(\{r, s\}, \Sigma, \delta', r, \{r, s\})$ defined by $\delta' = \{r, s\} \times \Sigma \times \{s\}$. Define $U' \colon \delta'^* \to \mathcal{F}_N^{\mathbb{K}}$ by:

$$U'((q, a_i, q'))(\overline{x}) = \begin{cases} U_{a_i}(\mathsf{Vec}(q_0, 0^d)) & \text{if } q = r \wedge q' = s, \\ U_{a_i}.\overline{x} & \text{otherwise, i.e., if } q = q' = s. \end{cases}$$

Finally, a special case should be added for the empty word: We let $C' = G$ if $\varepsilon \notin L(A, U, C)$ and $C' = G \cup \{0^N\}$ otherwise. We have that (A', U', C') is a $\mathbb{K}$-DetAPA where A' has only two states, and it is of the same language as (A, U, C). Finally, note that we need two states, and not one, because $\mathbb{K}$-APA use $\overline{0}$ as the starting value for their registers but $\overline{o}$ is needed here. ☐

We now give some negative properties of APA; our main tool is the following lemma:

Lemma 20. *Let L be a Turing-recognizable language. Then there exist effectively $L_1, L_2 \in \mathcal{L}_{\mathbb{Q}\text{-DetAPA}}$, and a morphism h such that $L = h(L_1 \cap L_2)$.*

Proof. This follows closely [1, Theorem 1], thus we only sketch the proof. Let M be a one-tape Turing machine, and suppose w.l.o.g. that M makes an odd number of steps on any accepting computation and that M only halts on accepting computation. Let L_1 be the set of strings

$$ID_0 \# ID_2 \# \cdots \# ID_{2k} \$ (ID_{2k+1})^R \# \cdots \# (ID_3)^R \# (ID_1)^R \tag{2}$$

such that the ID_i's are instantaneous descriptions of configurations of M, ID_0 is an initial configuration, ID_{2k+1} is an accepting configuration, and for all i, ID_{2i+1} is the configuration which would be reached in one step from configuration ID_{2i}. Similarly, L_2 is the same as L_1 but checks that ID_{2i} is the successor of ID_{2i-1}. These languages are in $\mathcal{L}_{\mathbb{Q}\text{-DetAPA}}$, using a technique similar to Proposition 17. Thus $L_1 \cap L_2$ is a language of $\mathcal{L}_{\mathbb{Q}\text{-DetAPA}}$ which encodes the strings of the type of 2 such that the ID_i's encode an accepting computation of M. Now if each string ID_i, $i > 0$, is over an alphabet which is disjoint from the alphabet which encodes the initial instantaneous description, then the morphism h which erases all of the symbols in a string of $L_1 \cap L_2$ except those representing the input is such that $L(M) = h(L_1 \cap L_2)$. ☐

Corollary 21. *Neither $\mathcal{L}_{\mathbb{K}\text{-APA}}$ nor $\mathcal{L}_{\mathbb{K}\text{-DetAPA}}$ is closed under morphisms.*

Corollary 22. *The emptiness problem is undecidable for DetAPA.*

Proof. Let $L \subseteq \Sigma^*$ be a Turing-recognizable language, and $x \in \Sigma^*$. Let L_1, L_2, h be given by Lemma 20 for L. Then $x \in L$ iff $L_1 \cap L_2 \cap h^{-1}(x)$ is nonempty, the latter being in $\mathcal{L}_{\mathbb{Q}\text{-DetAPA}}$. ☐

Recall that $\mathcal{L}_{\mathbb{K}\text{-APA}}$ is closed under concatenation. The previous property and the fact that a language L is empty iff $L \cdot \Sigma^*$ is finite implies:

Corollary 23. *Finiteness is undecidable for $\mathbb{K}$-APA.*

5. Parikh automata on letters

The PA *on letters* requires that the "weight" of a transition only depend on the input letter from Σ triggering the transition. In a way similar to the CA characterization of PA, we characterize PA on letters solely in terms of automata on Σ and semilinear sets. This model helps us in proving a standard lemma in language theory, in the context of PA.

Definition 5 (Parikh automaton on letters)**.** A *Parikh automaton on letters (LPA)* is a PA (A, C) where whenever $(a, \overline{v_1})$ and $(a, \overline{v_2})$ are labels of some transitions in A, then $\overline{v_1} = \overline{v_2}$. We write $\mathcal{L}_{\text{LPA}}$ (resp. $\mathcal{L}_{\text{DetLPA}}$) for the class of languages recognized by LPA (resp. LPA which are DetPA).

Now let (A, C) be a LPA. We may determinize A in the standard way and, although this is not the case with a PA, the resulting LPA is deterministic, thus:

Proposition 24. $\mathcal{L}_{\text{LPA}} = \mathcal{L}_{\text{DetLPA}}$.

For $R \subseteq \Sigma^*$ and $C \subseteq \mathbb{N}^{|\Sigma|}$, define $R\restriction_C = \{w \in R \mid \Phi(w) \in C\}$. Then:

Proposition 25. *Let $L \subseteq \Sigma^*$ be a language. The following are equivalent:*
(i) $L \in \mathcal{L}_{\text{LPA}}$;
(ii) There exist a regular language $R \subseteq \Sigma^*$ and a semilinear set $C \subseteq \mathbb{N}^{|\Sigma|}$ such that $R\restriction_C = L$.

The following property will be our central tool for showing nonclosure results:

Lemma 26. *Let $L \in \mathcal{L}_{\text{LPA}}$. For any regular language E:*

$$L \cap E \textit{ is not regular} \quad \Rightarrow \quad (\exists w \in E)[c(w) \cap L = \emptyset].$$

Proof. Let $R \subseteq \Sigma^*$ be a regular language and $C \subseteq \mathbb{N}^{|\Sigma|}$ be a semilinear set. Define $L = R\restriction_C$. Let E be a regular language such that $L \cap E$ is not regular. As $L \subseteq R$, we have $(L \cap E) \subseteq (R \cap E)$. The left hand side being non regular, those two sets differ. Thus, let $w \in (R \cap E)$ such that $w \notin L \cap E$, we have $w \notin L$. Hence, $w \in (R \setminus L)$, which implies that $\Phi(w) \notin C$, and in turn, $c(w) \cap L = \emptyset$. □

Remark. Lemma 26 holds with, e.g., "context-free" in lieu of "regular", but the version given will suffice for our purposes.

Proposition 27. *(1) $\mathcal{L}_{\text{LPA}}$ is not closed under union, complement, squaring, nonerasing morphisms;*
(2) $\mathcal{L}_{\text{LPA}}$ is closed under intersection, inverse morphisms, commutative closure.

Proof. (1). (Union.) Let $L_1 = \{w \in \{a, b\}^* \mid |w|_a = |w|_b\}$ and $L_2 = b(a \cup b)^*$ be two languages of LPA. Suppose $L = L_1 \cup L_2 \in \mathcal{L}_{\text{LPA}}$. Let E be the regular language (a^+b^+). By the pumping lemma, $L \cap E$ is not regular, thus Lemma 26 states there exists $w \in E$ such that $c(w) \cap L = \emptyset$. But $u = b^{|w|_b}a^{|w|_a} \in c(w)$ and $u \in L$, a contradiction.
(Complement.) Note that L is the complement in $\{a, b\}^*$ of $\{a^m b^n \mid m > 0 \wedge m \neq n\}$, which is the language of a LPA.
(Squaring.) Let $L = \{a^m b^n \mid m \neq n\} \in \mathcal{L}_{\text{LPA}}$. Suppose $L^2 \in \mathcal{L}_{\text{LPA}}$, and let $E = (a^+b^+)^2$. Again, $L \cap E$ is not regular, Lemma 26 implies there exists $w \in E$ such that $c(w) \cap E = \emptyset$. But $a^{|w|_a}b^0a^0b^{|w|_b} \in c(w) \cap L$, a contradiction.
(Nonerasing morphisms.) We simply note that L is the image of the language $\{a_1^m b_1^n a_2^r b_2^s \mid m \neq n \wedge r \neq s\}$ by the morphism $h(a_i) = a, h(b_i) = b$.

(2). The proofs for the first two properties follow the usual proofs for finite automata. Closure under the commutative closure operator follows from the proof of Proposition 9. ☐

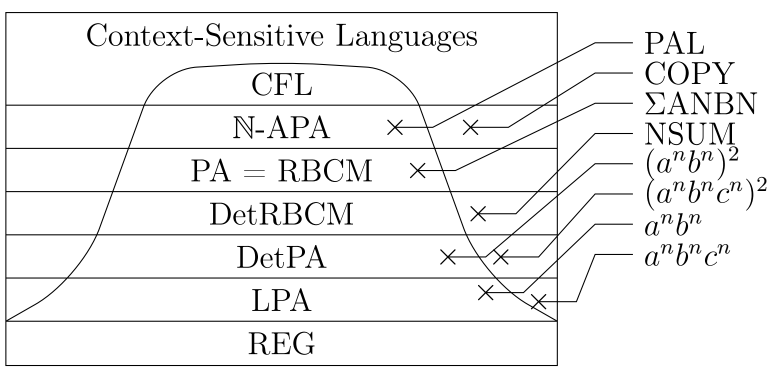
Finally, we use LPA to show the following property, which has a standard form known to be true for regular [19] and context-free languages [2] (the latter recently reworked in [8]). This property is sometimes called *Parikh-boundedness*:

Proposition 28. *For any $L \in \mathcal{L}_{\text{PA}}$, there exists a bounded language $L' \in \mathcal{L}_{\text{PA}}$ such that $L' \subseteq L$ and $\Phi(L) = \Phi(L')$.*

Proof. Let (A, C) be a constrained automaton, where δ is the transition set of A. Let $R \subseteq \delta^*$ and $D \subseteq \mathbb{N}^{|\delta|}$ be such that $\mu(R{\restriction_D}) = L(A, C)$. As mentioned, we can find a bounded regular language $R' \subseteq R$ such that $\Phi(R') = \Phi(R)$. In particular, $\Phi(R'{\restriction_D}) = \Phi(R{\restriction_D})$. Closure under morphism of $\mathcal{L}_{\text{PA}}$ implies that $L = \mu(R'{\restriction_D})$ is a bounded language of $\mathcal{L}_{\text{PA}}$ included in $L(A, C)$. Moreover, $\Phi(L(A, C)) = \Phi(\mu(R{\restriction_D}))$, and thus, equals $\Phi(L)$. ☐

6. Conclusion

The following table summarizes the current state of knowledge concerning the PA and its variants studied here; a class contains the class below it, and a language witnessing the separation is attached to the top class when we know this containment to be strict.



An intriguing question is whether there are context-free or context-sensitive languages outside $\mathcal{L}_{\mathbb{N}\text{-APA}}$. How difficult is that question? How about $\mathcal{L}_{\mathbb{N}\text{-DetAPA}}$? We have been unable to locate the latter class meaningfully. In particular, can $\mathcal{L}_{\mathbb{N}\text{-DetAPA}}$ be separated from $\mathcal{L}_{\mathbb{N}\text{-APA}}$?

The following summarizes the known closure and decidability properties for PA variants, and proposes open questions:

	$\cup$	$\cap$	$\cdot$	$\overline{}$	h	h^{-1}	c	$*$	$\emptyset$	Σ^*	fin.	$\subseteq$	reg.
LPA	N	Y	N	N	N	Y	Y	N	D	D	D	D	?
DetPA	Y	Y	N	Y	N	Y	Y	N	D	D	D	D	?
PA	Y	Y	Y	N	Y	Y	Y	N	D	U	U	U	U
DetAPA	Y	Y	?	Y	N	Y	?	?	U	U	?	U	?
APA	Y	Y	Y	?	N	Y	?	?	U	U	U	U	U

Several questions thus remain open concerning the poorly understood (and possibly overly powerful) affine PA model. But surely we expect testing a LPA or a DetPA for regularity to be decidable. How can regularity be tested for these models? One avenue for future research towards this goal might be characterizing $\mathcal{L}_{\mathrm{DetPA}}$ along the lines of algebraic automata theory.

Acknowledgments. The first author thanks L. Beaudou, M. Kaplan, and A. Lemaître.

References

[1] Baker, B.S., Book, R.V.: Reversal-bounded multipushdown machines. Journal of Computer and System Sciences 8(3), 315–332 (1974)
[2] Blattner, M., Latteux, M.: Parikh-bounded languages. In: ICALP. LNCS, vol. 115, pp. 316–323. Springer Berlin / Heidelberg (1981), 10.1007/3-540-10843-2_26
[3] Book, R., Nivat, M., Paterson, M.: Reversal-bounded acceptors and intersections of linear languages. SIAM Journal on Computing 3(4), 283 (1974)
[4] Brandenburg, F.: Analogies of PAL and COPY. In: Fundamentals of Computation Theory. Lecture Notes in Computer Science, vol. 117, pp. 61–70. Springer Berlin / Heidelberg (1981)
[5] Chiniforooshan, E., Daley, M., Ibarra, O.H., Kari, L., Seki, S.: One-reversal counter machines and multihead automata: revisited. In: SOFSEM. pp. 166–177 (2011), ACM ID: 1946384
[6] van den Dries, L.P.D.: Tame Topology and O-minimal Structures. Cambridge Univ. Press (1998)
[7] Enderton, H.B.: A Mathematical Introduction to Logic. Academic Press (1972)
[8] Ganty, P., Majumdar, R., Monmege, B.: Bounded underapproximations. In: Computer Aided Verification. p. 600–614 (2010)
[9] Ginsburg, S., Spanier, E.H.: Semigroups, Presburger formulas and languages. Pacific Journal of Mathematics 16(2), 285–296 (1966)
[10] Ginsburg, S., Spanier, E.: Finite-turn pushdown automata. SIAM Journal on Control and Optimization 4(3), 429 (1966)
[11] Greibach, S.A.: A note on undecidable properties of formal languages. Math Systems Theory 2(1), 1–6 (1968)
[12] Ibarra, O.H.: Reversal-bounded multicounter machines and their decision problems. J. ACM 25(1), 116–133 (1978)
[13] Ibarra, O.H., Su, J.: A technique for proving decidability of containment and equivalence of linear constraint queries. J. Comput. Syst. Sci. 59(1), 1–28 (1999)
[14] Karianto, W.: Parikh automata with pushdown stack. Diploma thesis, RWTH Aachen (2004)
[15] Klaedtke, F., Rueß, H.: Parikh automata and monadic second-order logics with linear cardinality constraints. Tech. rep. 177, Universität Freiburg (2002)
[16] Klaedtke, F., Rueß, H.: Monadic second-order logics with cardinalities. In: ICALP. LNCS, vol. 2719, pp. 681–696. Springer-Verlag (2003)
[17] Klarlund, N., Schneider, F.B.: Verifying safety properties using non-deterministic infinite-state automata. Tech. rep., Ithaca, NY, USA (1989)
[18] Kuroda, S.Y.: Classes of languages and linear bounded automata. Information and Control 7(2), 207–223 (1964)
[19] Latteux, M.: Mots infinis et langages commutatifs. RAIRO Info. Théo. 12(3), 185–192 (1978)
[20] Parikh, R.J.: On context-free languages. Journal of the ACM 13(4), 570–581 (1966)
[21] Straubing, H.: Finite Automata, Formal Logic, and Circuit Complexity. Birkhäuser, Boston (1994)
[22] Wolper, P., Boigelot, B.: An automata-theoretic approach to Presburger arithmetic constraints. In: Static Analysis (SAS'95). LNCS, vol. 983, pp. 21–32. Springer Berlin / Heidelberg (1995)