Remarks on Parikh-Recognizable Omega-languages

Mario Grobler
University of Bremen, Germany

Leif Sabellek
University of Bremen, Germany

Sebastian Siebertz
University of Bremen, Germany

Abstract

Several variants of Parikh automata on infinite words were recently introduced by Guha et al. [FSTTCS, 2022]. We show that one of these variants coincides with blind counter machine as introduced by Fernau and Stiebe [Fundamenta Informaticae, 2008]. Fernau and Stiebe showed that every ω-language recognized by a blind counter machine is of the form $\bigcup_i U_i V_i^\omega$ for Parikh recognizable languages U_i, V_i, but blind counter machines fall short of characterizing this class of ω-languages. They posed as an open problem to find a suitable automata-based characterization. We introduce several additional variants of Parikh automata on infinite words that yield automata characterizations of classes of ω-language of the form $\bigcup_i U_i V_i^\omega$ for all combinations of languages U_i, V_i being regular or Parikh-recognizable. When both U_i and V_i are regular, this coincides with Büchi's classical theorem. We study the effect of ε-transitions in all variants of Parikh automata and show that almost all of them admit ε-elimination. Finally we study the classical decision problems with applications to model checking.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects

Keywords and phrases Parikh automata, blind counter machines, infinite words, Büchi's theorem

Digital Object Identifier 10.4230/LIPIcs.CSL.2024.31

Related Version This paper subsumes the unpublished results of [19, 21].
Full Version: `https://arxiv.org/abs/2307.07238` [20]

Acknowledgements We thank Georg Zetzsche for his valuable remarks.

1 Introduction

Finite automata find numerous applications in formal language theory, logic, verification, and many more, in particular due to their good closure properties and algorithmic properties. To enrich this spectrum of applications even more, it has been a fruitful direction to add features to finite automata to capture also situations beyond the regular realm.

One such possible extension of finite automata with counting mechanisms has been introduced by Greibach in her study of blind and partially blind (one-way) multicounter machines [18]. Blind multicounter machines are generalized by weighted automata as introduced in [28]. Parikh automata (PA) were introduced by Klaedtke and Rueß in [26]. A PA is a non-deterministic finite automaton that is additionally equipped with a semi-linear set C, and every transition is equipped with a d-tuple of non-negative integers. Whenever an input word is read, d counters are initialized with the values 0 and every time a transition is used, the counters are incremented by the values in the tuple of the transition accordingly. An input word is accepted if the PA ends in an accepting state and additionally, the resulting d-tuple of counter values lies in C. Klaedtke and Rueß showed that PA are equivalent to weighted automata over the group $(\mathbb{Z}^k, +, \mathbf{0})$, and hence equivalent to Greibach's blind multicounter machines, as well as to reversal bounded multicounter machines [2, 24]. Recently it was shown that these models can be translated into each other using only logarithmic space [3]. In this work we call the class of languages recognized by any of these models *Parikh recognizable*. Klaedtke and Rueß [26] showed that the class of Parikh recognizable languages is precisely the class of languages definable in weak existential monadic second-order logic of one successor extended with linear cardinality constraints. The class of Parikh-recognizable languages contains all regular languages, but also many more, even languages that are not context-free, e. g., the language $\{a^n b^n c^n \mid n \in \mathbb{N}\}$. On the other hand, the language of palindromes is context-free, but not Parikh-recognizable. On finite words, blind counter automata, Parikh automata and related models have been investigated extensively, extending [18, 26] for example by affine PA and PA on letters [6, 7], bounded PA [8], two-way PA [16], PA with a pushdown stack [25] as well as a combination of both [11], history-deterministic PA [12], automata and grammars with valences [13, 23], and several algorithmic applications, e.g. in the context of path logics for querying graphs [15].

In the well-studied realm of verification of reactive systems, automata-related approaches provide a powerful framework to tackle important problems such as the model checking problem [1, 9, 10]. However, computations of systems are generally represented as infinite objects, as we often expect them to not terminate (but rather interact with the environment). Hence, automata processing infinite words are suited for these tasks. One common approach is the following: assume we are given a system, e.g. represented as a Kripke structure K, and a specification represented as an automaton $\mathcal{A}$ (or any formalism that can be translated into one) accepting all counterexamples. Then we can verify that the system has no bad computations by solving intersection-emptiness of K and $\mathcal{A}$. Yet again, the most basic model of Büchi automata (which recognize ω-regular languages) are quite limited in their expressiveness, although they have nice closure properties.

Let us consider two examples. In a three-user setting in an operating system we would like to ensure that none of the users gets a lot more resources than the other two. A corresponding specification of bad computations can be modeled via the ω-language $\{\alpha \in \{a, b, c\}^\omega \mid \text{ there are infinitely many prefixes } w \text{ of } \alpha \text{ with } |w|_a > |w|_b + |w|_c\}$, stating that one user gets more resources than the other two users combined infinitely often. As another example, consider a classical producer-consumer setting, where a producer continuously produces a good, and a consumer consumes these goods continuously. We can model this setting as an infinite word and ask that at no time the consumer has consumed more than the producer has produced at this time. Bad computations can be modeled via the ω-language $\{\alpha \in \{p, c\}^\omega \mid \text{there is a prefix } w \text{ of } \alpha \text{ with } |w|_c > |w|_p\}$. Such specifications are not ω-regular, as these require to "count arbitrarily". This motivates the study of blind-counter and Parikh automata on infinite words, which was initiated by Fernau and Stiebe [14]. Independently, Klaedte and Rueß proposed possible extensions of Parikh automata on infinite words. This line of research was recently picked up by Guha et al. [22].

Guha et al. [22] introduced *safety, reachability, Büchi- and co-Büchi Parikh automata*. These models provide natural generalization of studied automata models with Parikh conditions on infinite words. One shortcoming of safety, reachability and co-Büchi Parikh automata is that they do not generalize Büchi automata, that is, they cannot recognize all ω-regular languages. The non-emptiness problem, which is highly relevant for model checking applications, is undecidable for safety and co-Büchi Parikh automata. Furthermore, none of these models has ω-closure, meaning that for every model there is a Parikh-recognizable language (on finite words) L such that L^ω is not recognizable by any of these models. Guha et al. raised the question whether (appropriate variants of) Parikh automata on infinite words have the same expressive power as blind counter automata on infinite words.

Büchi's famous theorem states that ω-regular languages are characterized as languages of the form $\bigcup_i U_i V_i^\omega$, where the U_i and V_i are regular languages [4]. As a consequence of the theorem, many properties of ω-regular languages are inherited from regular languages. For example, the non-emptiness problem for Büchi automata can basically be solved by testing non-emptiness for nondeterministic finite automata. In their systematic study of blind counter automata, Fernau and Stiebe [14] considered the class $\mathcal{K}_*$, the class of ω-languages of the form $\bigcup_i U_i V_i^\omega$ for Parikh-recognizable languages U_i and V_i. They proved that the class of ω-languages recognizable by blind counter machines is a proper subset of the class $\mathcal{K}_*$. They posed as an open problem to provide automata models that capture classes of ω-languages of the form $\bigcup_i U_i V_i^\omega$ where U_i and V_i are described by a certain mechanism.

In this work we propose *reachability-regular Parikh automata*, *limit Parikh automata*, and *reset Parikh automata* as new automata models.

We pick up the question of Fernau and Stiebe [14] to consider classes of ω-languages of the form $\bigcup_i U_i V_i^\omega$ where U_i and V_i are described by a certain mechanism. We define the four classes $\mathcal{L}^\omega_{\mathsf{Reg},\mathsf{Reg}}$, $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{Reg}}$, $\mathcal{L}^\omega_{\mathsf{Reg},\mathsf{PA}}$ and $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{PA}}$ of ω-languages of the form $\bigcup_i U_i V_i^\omega$, where the U_i, V_i are regular or Parikh-recognizable languages of finite words, respectively. By Büchi's theorem the class $\mathcal{L}^\omega_{\mathsf{Reg},\mathsf{Reg}}$ is the class of ω-regular languages.

We show that the newly introduced reachability-regular Parikh automata, which are a small modification of reachability Parikh automata (as introduced by Guha et al. [22]) capture exactly the class $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{Reg}}$. This model turns out to be equivalent to limit Parikh automata. This model was hinted at in the concluding remarks of [26].

Fully resolving the classification of the above mentioned classes we introduce reset Parikh automata. In contrast to all other Parikh models, these are closed under the ω-operation, while maintaining all algorithmic properties of PA (in particular, non-emptiness is NP-complete and hence decidable). We show that the class of Reset-recognizable ω-languages is a strict superclass of $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{PA}}$. We show that appropriate graph-theoretic restrictions of reset Parikh automata exactly capture the classes $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{PA}}$ and $\mathcal{L}^\omega_{\mathsf{Reg},\mathsf{PA}}$, yielding the first automata characterizations for these classes.

The automata models introduced by Guha et al. [22] do not have ε-transitions, while blind counter machines have such transitions. Towards answering the question of Guha et al. we study the effect of ε-transitions in all Parikh automata models. We show that all models except safety and co-Büchi Parikh automata admit ε-elimination. This in particular answers the question of Guha et al. [22] whether blind counter automata and Büchi Parikh automata have the same expressive power over infinite words affirmative. We show that safety and co-Büchi automata with ε-transitions are strictly more powerful than their variants without ε-transitions, and in particular, they give the models enough power to recognize all ω-regular languages.

All lemmas with missing proofs are marked with ($\star$), the full version [20] containing all proofs can be found on arXiv.

2 Preliminaries

2.1 Finite and infinite words

We write $\mathbb{N}$ for the set of non-negative integers including 0, and $\mathbb{Z}$ for the set of all integers. Let Σ be an alphabet, i.e., a finite non-empty set and let Σ^* be the set of all finite words over Σ. For a word $w \in \Sigma^*$, we denote by $|w|$ the length of w, and by $|w|_a$ the number of occurrences of the letter $a \in \Sigma$ in w. We write ε for the empty word of length 0.

An *infinite word* over an alphabet Σ is a function $\alpha : \mathbb{N} \setminus \{0\} \to \Sigma$. We often write α_i instead of $\alpha(i)$. Thus, we can understand an infinite word as an infinite sequence of symbols $\alpha = \alpha_1\alpha_2\alpha_3 \dots$ For $m \leq n$, we abbreviate the finite infix $\alpha_m \dots \alpha_n$ by $\alpha[m, n]$. We denote by Σ^ω the set of all infinite words over Σ. We call a subset $L \subseteq \Sigma^\omega$ an *ω-language*. Moreover, for $L \subseteq \Sigma^*$, we define $L^\omega = \{w_1w_2 \cdots \mid w_i \in L \setminus \{\varepsilon\}\} \subseteq \Sigma^\omega$.

2.2 Regular and ω-regular languages

A *nondeterministic finite automaton* (NFA) is a tuple $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$, where Q is the finite set of states, Σ is the input alphabet, $q_0 \in Q$ is the initial state, $\Delta \subseteq Q \times \Sigma \times Q$ is the set of transitions and $F \subseteq Q$ is the set of accepting states. A *run* of $\mathcal{A}$ on a word $w = w_1 \dots w_n \in \Sigma^*$ is a (possibly empty) sequence of transitions $r = r_1 \dots r_n$ with $r_i = (p_{i-1}, w_i, p_i) \in \Delta$ such that $p_0 = q_0$. We say r is *accepting* if $p_n \in F$. The empty run on ε is accepting if $q_0 \in F$. We define the *language recognized by* $\mathcal{A}$ as $L(\mathcal{A}) = \{w \in \Sigma^* \mid$ there is an accepting run of $\mathcal{A}$ on $w\}$. If a language L is recognized by some NFA $\mathcal{A}$, we call L *regular*.

A *Büchi automaton* is an NFA $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ that takes infinite words as input. A *run* of $\mathcal{A}$ on an infinite word $\alpha_1\alpha_2\alpha_3 \dots$ is an infinite sequence of transitions $r = r_1r_2r_3 \dots$ with $r_i = (p_{i-1}, \alpha_i, p_i) \in \Delta$ such that $p_0 = q_0$. We say r is *accepting* if there are infinitely many i with $p_i \in F$. We define the *ω-language recognized by* $\mathcal{A}$ as $L_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid$ there is an accepting run of $\mathcal{A}$ on $\alpha\}$. If an ω-language L is recognized by some Büchi automaton $\mathcal{A}$, we call L *ω-regular*. Büchi's theorem establishes an important connection between regular and ω-regular languages:

▶ **Theorem 1** (Büchi [4]). *A language $L \subseteq \Sigma^\omega$ is ω-regular if and only if there are regular languages $U_1, V_1, \dots, U_n, V_n \subseteq \Sigma^*$ for some $n \geq 1$ such that $L = U_1V_1^\omega \cup \dots \cup U_nV_n^\omega$.*

If every state of a Büchi automaton $\mathcal{A}$ is accepting, we call $\mathcal{A}$ a *safety automaton*.

2.3 Semi-linear sets

For some $d \geq 1$, a *linear set* of dimension d is a set of the form $\{b_0 + b_1z_1 + \dots + b_\ell z_\ell \mid z_1, \dots, z_\ell \in \mathbb{N}\} \subseteq \mathbb{N}^d$ for $b_0, \dots, b_\ell \in \mathbb{N}^d$. If $b_0 = \mathbf{0}$, then we call C a *homogeneous* linear set. A *semi-linear set* is a finite union of linear sets. For vectors $\mathbf{u} = (u_1, \dots, u_c) \in \mathbb{N}^c$ and $\mathbf{v} = (v_1, \dots, v_d) \in \mathbb{N}^d$, we denote by $\mathbf{u} \cdot \mathbf{v} = (u_1, \dots, u_c, v_1, \dots, v_d) \in \mathbb{N}^{c+d}$ the *concatenation of* $\mathbf{u}$ *and* $\mathbf{v}$. We extend this definition to sets of vectors. Let $C \subseteq \mathbb{N}^c$ and $D \subseteq \mathbb{N}^d$. Then $C \cdot D = \{\mathbf{u} \cdot \mathbf{v} \mid \mathbf{u} \in C, \mathbf{v} \in D\} \subseteq \mathbb{N}^{c+d}$. We denote by $\mathbf{0}^d$ (or simply $\mathbf{0}$ if d is clear from the context) the all-zero vector, and by $\mathbf{e}_i^d$ (or simply $\mathbf{e}_i$) the d-dimensional vector where the ith entry is 1 and all other entries are 0. We also consider semi-linear sets over $(\mathbb{N} \cup \{\infty\})^d$, that is semi-linear sets with an additional symbol ∞ for infinity. As usual, addition of vectors and multiplication of a vector with a number is defined component-wise, where $z + \infty = \infty + z = \infty + \infty = \infty$ for all $z \in \mathbb{N}$, $z \cdot \infty = \infty \cdot z = \infty$ for all $z > 0 \in \mathbb{N}$, and $0 \cdot \infty = \infty \cdot 0 = 0$.

2.4 Parikh-recognizable languages

A *Parikh automaton* (PA) is a tuple $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ where Q, Σ, q_0, and F are defined as for NFA, $\Delta \subseteq Q \times \Sigma \times \mathbb{N}^d \times Q$ is a finite set of *labeled transitions*, and $C \subseteq \mathbb{N}^d$ is a semi-linear set. We call d the *dimension* of $\mathcal{A}$ and refer to the entries of a vector $\mathbf{v}$ in a transition $(p, a, \mathbf{v}, q)$ as *counters*. Similar to NFA, a *run* of $\mathcal{A}$ on a word $w = x_1 \dots x_n$ is a (possibly empty) sequence of labeled transitions $r = r_1 \dots r_n$ with $r_i = (p_{i-1}, x_i, \mathbf{v}_i, p_i) \in \Delta$

such that $p_0 = q_0$. We define the *extended Parikh image* of a run r as $\rho(r) = \sum_{i \leq n} \mathbf{v}_i$ (with the convention that the empty sum equals $\mathbf{0}$). We say r is accepting if $p_n \in F$ and $\rho(r) \in C$, referring to the latter condition as the *Parikh condition*. We define the *language recognized by* $\mathcal{A}$ as $L(\mathcal{A}) = \{w \in \Sigma^* \mid \text{there is an accepting run of } \mathcal{A} \text{ on } w\}$. If a language $L \subseteq \Sigma^*$ is recognized by some PA, then we call L *Parikh-recognizable*.

2.5 Graphs

A *(directed) graph* G consists of its vertex set $V(G)$ and edge set $E(G) \subseteq V(G) \times V(G)$. In particular, a graph G may have loops, that is, edges of the form (u, u). A *(simple) path* from a vertex u to a vertex v in G is a sequence of pairwise distinct vertices $v_1 \dots v_k$ such that $v_1 = u$, $v_k = v$, and $(v_i, v_{i+1}) \in E(G)$ for all $1 \leq i < k$. Similarly, a *(simple) cycle* in G is a sequence of pairwise distinct vertices $v_1 \dots v_k$ such that $(v_i, v_{i+1}) \in E(G)$ for all $1 \leq i < k$, and $(v_k, v_1) \in E(G)$. If G has no cylces, we call G a directed acyclic graph (DAG). For a subset $U \subseteq V(G)$, we denote by $G[U]$ the graph G *induced by* U, i. e., the graph with vertex set U and edge set $\{(u, v) \in E(G) \mid u, v \in U\}$. A *strongly connected component* (SCC) in G is a maximal subset $U \subseteq V(G)$ such that for all $u, v \in U$ there is a path from u to v, i. e., all vertices in U are reachable from each other. We write $SCC(G)$ for the set of all strongly connected components of G (observe that $SCC(G)$ partitions $V(G)$). The *condensation* of G, written $C(G)$, is the DAG obtained from G by contracting each SCC of G into a single vertex, that is $V(C(G)) = SCC(G)$ and $(U, V) \in E(C(G))$ if and only if there is $u \in U$ and $v \in V$ with $(u, v) \in E(G)$. We call the SCCs with no outgoing edges in $C(G)$ leaves. Note that an automaton can be seen as a labeled graph. Hence, all definitions translate to automata by considering the underlying graph (to be precise, an automaton can be seen as a labeled multigraph; however, we simply drop parallel edges).

3 Parikh automata on infinite words

In this section, we recall the acceptance conditions of Parikh automata operating on infinite words that were studied before in the literature and introduce our new models. We make some easy observations and compare the existing with the new automata models. We define only the non-deterministic variants of these automata.

Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ be a PA. A run of $\mathcal{A}$ on an infinite word $\alpha = \alpha_1\alpha_2\alpha_3 \dots$ is an infinite sequence of labeled transitions $r = r_1r_2r_3 \dots$ with $r_i = (p_{i-1}, \alpha_i, \mathbf{v}_i, p_i) \in \Delta$ such that $p_0 = q_0$. The automata defined below differ only in their acceptance conditions. In the following, whenever we say that an automaton $\mathcal{A}$ accepts an infinite word α, we mean that there is an accepting run of $\mathcal{A}$ on α.

1. The run r satisfies the *safety condition* if for every $i \geq 0$ we have $p_i \in F$ and $\rho(r_1 \dots r_i) \in C$. We call a PA accepting with the safety condition a *safety PA* [22]. We define the ω-language recognized by a safety PA $\mathcal{A}$ as $S_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}$.
2. The run r satisfies the *reachability condition* if there is an $i \geq 1$ such that $p_i \in F$ and $\rho(r_1 \dots r_i) \in C$. We say there is an *accepting hit* in r_i. We call a PA accepting with the reachability condition a *reachability PA* [22]. We define the ω-language recognized by a reachability PA $\mathcal{A}$ as $R_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}$.
3. The run r satisfies the *Büchi condition* if there are infinitely many $i \geq 1$ such that $p_i \in F$ and $\rho(r_1 \dots r_i) \in C$. We call a PA accepting with the Büchi condition a *Büchi PA* [22]. We define the ω-language recognized by a Büchi PA $\mathcal{A}$ as $B_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}$. Hence, a Büchi PA can be seen as a stronger variant of a reachability PA where we require infinitely many accepting hits instead of a single one.

4. The run r satisfies the *co-Büchi condition* if there is i_0 such that for every $i \geq i_0$ we have $p_i \in F$ and $\rho(r_1 \dots r_i) \in C$. We call a PA accepting with the co-Büchi condition a *co-Büchi PA* [22]. We define the ω-language recognized by a co-Büchi PA $\mathcal{A}$ as $CB_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}$.
   Hence, a co-Büchi PA can be seen as a weaker variant of safety PA where the safety condition needs not necessarily be fulfilled from the beginning, but from some point onwards.

Guha et al. [22] assume that reachability PA are complete, i.e., for every $(p, a) \in Q \times \Sigma$ there are $\mathbf{v} \in \mathbb{N}^d$ and $q \in Q$ such that $(p, a, \mathbf{v}, q) \in \Delta$, as incompleteness allows to express additional safety conditions. We also make this assumption in order to study "pure" reachability PA. In fact, we can assume that all models are complete, as the other models can be completed by adding a non-accepting sink. We remark that Guha et al. also considered *asynchronous* reachability and Büchi PA, where the Parikh condition does not necessarily need to be satisfied in accepting states. However, for non-deterministic automata this does not change the expressiveness of the considered models [22].

We now define the models newly introduced in this work. As already observed in [22] among the above considered models only Büchi PA can recognize all ω-regular languages. For example, $\{\alpha \in \{a, b\}^\omega \mid |\alpha|_a = \infty\}$ cannot be recognized by safety PA, reachability PA or co-Büchi PA.

We first extend reachability PA with the classical Büchi condition to obtain *reachability-regular PA*. In Theorem 9 we show that these automata characterize ω-languages of the form $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{Reg}}$, hence, providing a robust and natural model.

5. The run satisfies the *reachability and regularity condition* if there is an $i \geq 1$ such that $p_i \in F$ and $\rho(r_1 \dots r_i) \in C$, and there are infinitely many $j \geq 1$ such that $p_j \in F$. We call a PA accepting with the reachability and regularity condition a *reachability-regular* PA. We define the ω-language recognized by a reachability-regular PA $\mathcal{A}$ as $RR_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}$ and call it reachability-regular.

Note that (in contrast to reachability PA) we may assume that reachability-regular PA are complete without changing their expressiveness. Observe that every ω-regular language is reachability-regular, as we can turn an arbitrary Büchi automaton into an equivalent reachability-regular PA by labeling every transition with 0 and setting $C = \{0\}$.

We next introduce *limit PA*, which were proposed in the concluding remarks of [26]. As we will prove in Theorem 9, this seemingly quite different model is equivalent to reachability-regular PA.

6. The run satisfies the *limit condition* if there are infinitely many $i \geq 1$ such that $p_i \in F$, and if additionally $\rho(r) \in C$, where the jth component of $\rho(r)$ is computed as follows. If there are infinitely many $i \geq 1$ such that the jth component of $\mathbf{v}_i$ has a non-zero value, then the jth component of $\rho(r)$ is ∞. In other words, if the sum of values in a component diverges, then its value is set to ∞. Otherwise, the infinite sum yields a positive integer. We call a PA accepting with the limit condition a *limit PA*. We define the ω-language recognized by a limit PA $\mathcal{A}$ as $L_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}$.

Still, none of the yet introduced models have ω-closure. This shortcoming is addressed with the following two models, which will turn out to be equivalent and form the basis of the automata characterization of $\mathcal{L}^\omega_{\mathsf{Reg},\mathsf{PA}}$ and $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{PA}}$.

7. The run satisfies the *strong reset condition* if the following holds. Let $k_0 = 0$ and denote by $k_1 < k_2 < \dots$ the positions of all accepting states in r. Then r is accepting if $k_1, k_2, \dots$ is an infinite sequence and $\rho(r_{k_{i-1}+1} \dots r_{k_i}) \in C$ for all $i \geq 1$. We call a PA accepting with the strong reset condition a *strong reset PA*. We define the ω-language recognized by a strong reset PA $\mathcal{A}$ as $SR_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}$.
8. The run satisfies the *weak reset condition* if there are infinitely many reset positions $0 = k_0 < k_1 < k_2, \dots$ such that $p_{k_i} \in F$ and $\rho(r_{k_{i-1}+1} \dots r_{k_i}) \in C$ for all $i \geq 1$. We call a PA accepting with the weak reset condition a *weak reset PA*. We define the ω-language recognized by a weak reset PA $\mathcal{A}$ as $WR_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}$.

Intuitively worded, whenever a strong reset PA enters an accepting state, the Parikh condition *must* be satisfied. Then the counters are reset. Similarly, a weak reset PA may reset the counters whenever there is an accepting hit, and they must reset infinitely often, too. In the following we will often just speak of reset PA without explicitly stating whether they are weak or strong. In this case, we mean the strong variant. We will show the equivalence of the two models in Lemma 26 and Lemma 27.

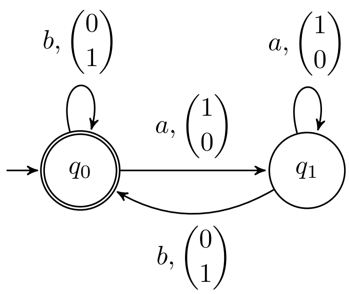


Figure 1 The automaton $\mathcal{A}$ with $C = \{(z, z'), (z, \infty) \mid z' \geq z\}$ from Example 1.

▶ **Example 1.** Let $\mathcal{A}$ be the automaton in Figure 1 with $C = \{(z, z'), (z, \infty) \mid z' \geq z\}$.

- If we interpret $\mathcal{A}$ as a PA (over finite words), then we have $L(\mathcal{A}) = \{w \in \{a, b\}^* \cdot \{b\} \mid |w|_a \leq |w|_b\} \cup \{\varepsilon\}$. The automaton is in the accepting state at the very beginning and every time after reading a b. The first counter counts the occurrences of letter a, the second one counts occurrences of b. By definition of C the automaton only accepts when the second counter value is greater or equal to the first counter value (note that vectors containing an ∞-entry have no additional effect).
- If we interpret $\mathcal{A}$ as a safety PA, then we have $S_\omega(\mathcal{A}) = \{b\}^\omega$. As q_1 is not accepting, only the b-loop on q_0 may be used.
- If we interpret $\mathcal{A}$ as a reachability PA, then we have $R_\omega(\mathcal{A}) = \{\alpha \in \{a, b\}^\omega \mid \alpha \text{ has a prefix in } L(\mathcal{A})\}$. The automaton has satisfied the reachability condition after reading a prefix in $L(\mathcal{A})$ and accepts any continuation after that.
- If we interpret $\mathcal{A}$ as a Büchi PA, then we have $B_\omega(\mathcal{A}) = L(\mathcal{A})^\omega$. The automaton accepts an infinite word if infinitely often the Parikh condition is satisfied in the accepting state. Observe that C is a homogeneous linear set and the initial state as well as the accepting state have the same outgoing transitions.
- If we interpret $\mathcal{A}$ as a co-Büchi PA, then we have $CB_\omega(\mathcal{A}) = L(\mathcal{A}) \cdot \{b\}^\omega$. This is similar to the safety PA, but the accepted words may have a finite "non-safe" prefix from $L(\mathcal{A})$.

- If we interpret $\mathcal{A}$ as a reachability-regular PA, then we have $RR_\omega(\mathcal{A}) = \{\alpha \in \{a,b\}^\omega \mid \alpha$ has a prefix in $L(\mathcal{A})$ and $|\alpha|_b = \infty\}$. After having met the reachability condition the automaton still needs to satisfy the Büchi condition, which enforces infinitely many visits of the accepting state.
- If we interpret $\mathcal{A}$ as a limit PA, then we have $L_\omega(\mathcal{A}) = \{\alpha \in \{a,b\}^\omega \mid |\alpha|_a < \infty\}$. The automaton must visit the accepting state infinitely often. At the same time the extended Parikh image must belong to C, which implies that the infinite word contains only some finite number z of letter a (note that only the vectors of the form (z,∞) have an effect here, as at least one symbol must be seen infinitely often by the infinite pigeonhole principle).
- If we interpret $\mathcal{A}$ as a weak reset PA, then we have $WR_\omega(\mathcal{A}) = L(\mathcal{A})^\omega$. As a weak reset PA may (but is not forced to) reset the counters upon visiting the accepting state, the automaton may reset every time a (finite) infix in $L(\mathcal{A})$ has been read.
- If we interpret $\mathcal{A}$ as a strong reset PA, then we have $SR_\omega(\mathcal{A}) = \{b^*a\}^\omega \cup \{b^*a\}^* \cdot \{b\}^\omega$. Whenever the automaton reaches an accepting state also the Parikh condition must be satisfied. This implies that the a-loop on q_1 may never be used, as this would increase the first counter value to at least 2, while the second counter value is 1 upon reaching the accepting state q_0 (which resets the counters).

▶ Remark. The automaton $\mathcal{A}$ in the example is deterministic. We note that $L_\omega(\mathcal{A})$ is not deterministic ω-regular but deterministic limit PA-recognizable.

4 Büchi-like characterizations

It was observed in [22] that Büchi PA recognize a strict subset of $\mathcal{L}^\omega_{\mathsf{PA,PA}}$. In this section we first show that the class of reset PA-recognizable ω-languages is a strict superset of $\mathcal{L}^\omega_{\mathsf{PA,PA}}$. Then we provide an automata-based characterization of $\mathcal{L}^\omega_{\mathsf{PA,Reg}}$, $\mathcal{L}^\omega_{\mathsf{PA,PA}}$, and $\mathcal{L}^\omega_{\mathsf{Reg,PA}}$. Towards this goal we first establish some closure properties.

Guha et al. [22] have shown that safety, reachability, Büchi, and co-Büchi PA are closed under union using a modification of the standard construction for PA, i. e., taking the disjoint union of the automata (introducing a fresh initial state), and the disjoint union of the semi-linear sets, where disjointness is achieved by "marking" every vector in the first set by an additional 1 (increasing the dimension by 1), and all vectors in the second set by an additional 2. We observe that the same construction also works for reachability-regular and limit PA, and a small modification is sufficient to make the construction also work for reset PA. We leave the details to the reader.

▶ **Lemma 2.** *The classes of reachability-regular, limit PA-recognizable, and reset PA-recognizable ω-languages are closed under union.*

Furthermore, we show that these classes, as well as the class of Büchi PA-recognizable ω-languages, are closed under left-concatenation with PA-recognizable languages. We provide some details in the next lemma, as we will need to modify the standard construction in such a way that we do not need to keep accepting states of the PA on finite words. This will help to characterize $\mathcal{L}^\omega_{\mathsf{PA,PA}}$ via (restricted) reset PA.

▶ **Lemma 3** ($\star$)**.** *The classes of reachability-regular, limit PA-recognizable, reset PA-recognizable, and Büchi PA-recognizable ω-languages are closed under left-concatenation with PA-recognizable languages.*

Before we continue, we show that we can normalize PA (on finite words) such that the initial state is the only accepting state. This observation simplifies several proofs in this section.

▶ **Lemma 4** ($\star$). *Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ be a PA of dimension d. Then there exists an equivalent PA $\mathcal{A}'$ of dimension $d+1$ with the following properties.*
- *The initial state of $\mathcal{A}'$ is the only accepting state.*
- *$SCC(\mathcal{A}') = \{Q\}$.*

We say that $\mathcal{A}'$ is normalized.

Observe that we have $SR_\omega(\mathcal{A}') = L(\mathcal{A})^\omega$, that is, every normalized PA interpreted as a reset PA recognizes the ω-closure of the language recognized by the PA. As an immediate consequence we obtain the following corollary.

▶ **Corollary 5.** *The class of reset PA-recognizable ω-languages is closed under the ω-operation.*

Combining these results we obtain that every ω-language in $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{PA}}$, i.e. every ω-language of the form $\bigcup_i U_i V_i^\omega$ is reset PA-recognizable. We show that the other direction does not hold, i.e., the inclusion is strict.

▶ **Lemma 6.** *The class $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{PA}}$ is a strict subclass of the class of reset PA-recognizable ω-languages.*

Proof. The inclusion is a direct consequence of Lemma 2, Lemma 3, and Corollary 5. Hence we show that the inclusion is strict.

Consider the ω-language $L = \{a^n b^n \mid n \geq 1\}^\omega \cup \{a^n b^n \mid n \geq 1\}^* \cdot \{a\}^\omega$. This ω-language is reset PA-recognizable, as witnessed by the strong reset PA in Figure 2 with $C = \{(z, z) \mid z \in \mathbb{N}\}$.

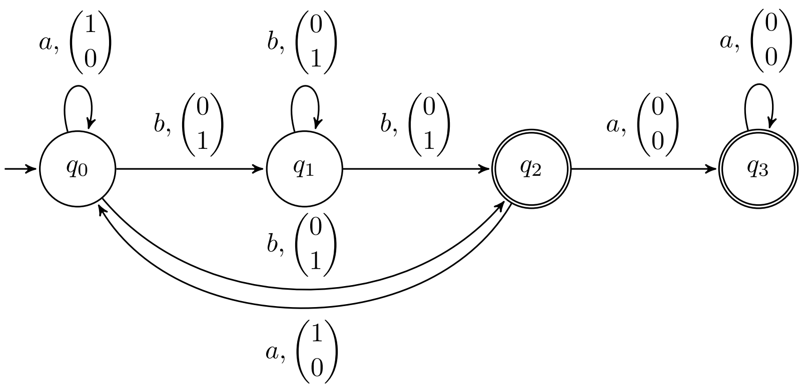


Figure 2 The strong reset PA for $L = \{a^n b^n \mid n \geq 1\}^\omega \cup \{a^n b^n \mid n \geq 1\}^* \cdot \{a\}^\omega$.

We claim that $L \notin \mathcal{L}^\omega_{\mathsf{PA},\mathsf{PA}}$. Assume towards a contraction that $L \in \mathcal{L}^\omega_{\mathsf{PA},\mathsf{PA}}$, i. e., there are Parikh-recognizable languages $U_1, V_1, \ldots, U_n, V_n$ such that $L = U_1 V_1^\omega \cup \cdots \cup U_n V_n^\omega$. Then there is some $i \leq n$ such that for infinitely many $j \geq 1$ the infinite word $\alpha_j = aba^2b^2 \ldots a^j b^j \cdot a^\omega \in U_i V_i^\omega$. Then V_i must contain a word of the form $v = a^k$, $k > 0$. Additionally, there cannot be a word in V_i with infix b. To see this assume for sake of contradiction that there is a word $w \in V_i$ with $\ell = |w|_b > 0$. Let $\beta = (v^{\ell+1} w)^\omega$. Observe that β has an infix that consists of at least $\ell + 1$ many a, followed by at most ℓ, but at least one b, hence, no word of the form $u\beta$ with $u \in U_i$ is in L. This is a contradiction, thus $V_i \subseteq \{a\}^+$.

Since $U_i \in \mathcal{L}_{\mathsf{PA}}$, there is a PA $\mathcal{A}_i$ with $L(\mathcal{A}_i) = U_i$. Let m be the number of states in $\mathcal{A}_i$ and $w' = aba^2b^2 \dots a^{m^4+1}b^{m^4+1}$. Then w' is a prefix of a word accepted by $\mathcal{A}_i$. Now consider the infixes $a^\ell b^\ell$ and the pairs of states q_1, q_2, where we start reading a^ℓ and end reading a^ℓ, and q_3, q_4 where we start to read b^ℓ and end to read b^ℓ, respectively. There are m^2 choices for the first pair and m^2 choices for the second pair, hence m^4 possibilities in total. Hence, as we have more than m^4 such infixes, there must be two with the same associated states q_1, q_2, q_3, q_4. Then we can swap these two infixes and get a word of the form $ab \dots a^r b^s \dots a^s b^r \dots a^{m^4+1}b^{m^4+1}$ that is a prefix of some word in $L(\mathcal{A}_i) = U_i$. But no word in L has such a prefix, a contradiction. Thus, $U_1V_1^\omega \cup \dots \cup U_nV_n^\omega \neq L$. ◀

4.1 Characterization of Büchi Parikh automata

As mentioned in the last section, the class of ω-languages recognized by Büchi PA is a strict subset of $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{PA}}$, i. e., languages of the form $\bigcup_i U_iV_i^\omega$ for Parikh-recognizable U_i and V_i. In this subsection we show that a restriction of the PA recognizing the V_i is sufficient to exactly capture the expressiveness of Büchi PA. To be precise, we show the following.

▶ **Lemma 7.** *The following are equivalent for all ω-languages $L \subseteq \Sigma^\omega$:*

1. *L is Büchi PA-recognizable.*
2. *L is of the form $\bigcup_i U_iV_i^\omega$, where $U_i \in \Sigma^*$ is Parikh-recognizable and $V_i \in \Sigma^*$ is recognized by a normalized PA where C is a homogeneous linear set.*

We note that we can translate every PA (with a linear set C) into an equivalent normalized PA by Lemma 4. However, this construction adds a base vector, as we concatenate $\{1\}$ to C. In fact, this can generally not be avoided without losing expressiveness. It turns out that this loss of expressiveness is exactly what we need to characterize the class of ω-languages recognized by Büchi PA as stated in the previous lemma. The main reason for this is pointed out in the following lemma.

▶ **Lemma 8** ($\star$)**.** *Let L be a language recognized by a (normalized) PA $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \{q_0\}, C)$ where C is a homogeneous linear set. Then we have $B_\omega(\mathcal{A}) = L(\mathcal{A})^\omega$.*

This is the main ingredient to prove Lemma 7.

Proof of Lemma 7. We note that the proof in [22] showing that every ω-language L recognized by a Büchi-PA is of the form $\bigcup_i U_iV_i$ for PA-recognizable U_i and V_i already constructs PA for the V_i of the desired form. This shows the implication (1) $\Rightarrow$ (2).

To show the implication (2) $\Rightarrow$ (1), we use that the ω-closure of languages recognized by PA of the stated form is Büchi PA-recognizable by Lemma 8. As Büchi PA are closed under left-concatenation with PA-recognizable languages (Lemma 3) and union [22], the claim follows. ◀

4.2 Characterization of $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{Reg}}$

In this subsection we characterize $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{Reg}}$ by showing the following equivalences.

▶ **Theorem 9.** *The following are equivalent for all ω-languages $L \subseteq \Sigma^\omega$.*

1. *L is of the form $\bigcup_i U_iV_i^\omega$, where $U_i \in \Sigma^*$ is Parikh-recognizable, and $V_i \subseteq \Sigma^*$ is regular.*
2. *L is limit PA-recognizable.*
3. *L is reachability-regular.*

Observe that in the first item we may assume that L is of the form $\bigcup_i U_i V_i$, where $U_i \in \Sigma^*$ is Parikh-recognizable, and $V_i \subseteq \Sigma^\omega$ is ω-regular. Then, by simple combinatorics and Büchi's theorem we have $\bigcup_i U_i V_i = \bigcup_i U_i(\bigcup_{j_i} X_{j_i} Y_{j_i}^\omega) = \bigcup_{i,j_i} U_i(X_{j_i} Y_{j_i}^\omega) = \bigcup_{i,j_i} (U_i X_{j_i}) Y_{j_i}^\omega$, for regular languages X_{j_i}, Y_{j_i}, where $U_i X_{j_i}$ is Parikh-recognizable, as Parikh-recognizable languages are closed under concatenation [5, Proposition 3].

To simplify the proof, it is convenient to consider the following generalizations of Büchi automata. A *transition-based generalized Büchi automaton* (TGBA) is a tuple $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{T})$ where $\mathcal{T} \subseteq 2^\Delta$ is a collection of sets of transitions. Then a run $r_1 r_2 r_3 \dots$ of $\mathcal{A}$ is accepting if for all $T \in \mathcal{T}$ there are infinitely many i such that $r_i \in T$. It is well-known that TGBA have the same expressiveness as Büchi automata [17].

Theorem 9 will be a direct consequence from the following lemmas. The first lemma shows the implication (1) $\Rightarrow$ (2).

▶ **Lemma 10.** *If $L \in \mathcal{L}^\omega_{\mathsf{PA},\mathsf{Reg}}$, then L is limit PA-recognizable.*

Proof. As the class of limit PA-recognizable ω-languages is closed under union by Lemma 2, it is sufficient to show how to construct a limit PA for an ω-language of the form $L = UV^\omega$, where U is Parikh-recognizable and V is regular.

Let $\mathcal{A}_1 = (Q_1, \Sigma, q_1, \Delta_1, F_1, C)$ be a PA with $L(\mathcal{A}_1) = U$ and $\mathcal{A}_2 = (Q_2, \Sigma, q_2, \Delta_2, F_2)$ be a Büchi automaton with $L_\omega(\mathcal{A}_2) = V^\omega$. We use the following standard construction for concatenation. Let $\mathcal{A} = (Q_1 \cup Q_2, \Sigma, q_1, \Delta, F_2, C)$ be a limit PA where

$$\Delta = \Delta_1 \cup \{(p, a, \mathbf{0}, q) \mid (p, a, q) \in \Delta_2\} \cup \{(f, a, \mathbf{0}, q) \mid (q_2, a, q) \in \Delta_2, f \in F_1\}.$$

We claim that $L_\omega(\mathcal{A}) = L$.

$\Rightarrow$ To show $L_\omega(\mathcal{A}) \subseteq L$, let $\alpha \in L_\omega(\mathcal{A})$ with accepting run $r_1 r_2 r_3 \dots$ where $r_i = (p_{i-1}, \alpha_i, \mathbf{v}_i, p_i)$. As only the states in F_2 are accepting, there is a position j such that $p_{j-1} \in F_1$ and $p_j \in Q_2$. In particular, all transitions of the copy of $\mathcal{A}_2$ are labeled with $\mathbf{0}$, i. e., $\mathbf{v}_i = \mathbf{0}$ for all $i \geq j$. Hence $\rho(r) = \rho(r_1 \dots r_{j-1}) \in C$ (in particular, there is no ∞ value in $\rho(r)$). We observe that $r_1 \dots r_{j-1}$ is an accepting run of $\mathcal{A}_1$ on $\alpha[1, j-1]$, as $p_{j-1} \in F_1$ and $\rho(r_1 \dots r_{j-1}) \in C$. For all $i \geq j$ let $r'_i = (p_{i-1}, \alpha_i, p_i)$. Observe that $(q_2, \alpha_j, p_j) r'_{j+1} r'_{j+2} \dots$ is an accepting run of $\mathcal{A}_2$ on $\alpha_j \alpha_{j+1} \alpha_{j+2} \dots$, hence $\alpha \in L(\mathcal{A}_1) \cdot L_\omega(\mathcal{A}_2) = L$.

$\Leftarrow$ To show $L = UV^\omega \subseteq L_\omega(\mathcal{A})$, let $w \in L(\mathcal{A}_1) = U$ with accepting run s, and $\alpha \in L_\omega(\mathcal{A}_2) = V^\omega$ with accepting run $r = r_1 r_2 r_3 \dots$, where $r_i = (p_{i-1}, \alpha_1, p_i)$. Observe that s is also a partial run of $\mathcal{A}$ on w, ending in an accepting state f. By definition of Δ, we can continue the run s in $\mathcal{A}$ basically as in r. To be precise, let $r'_1 = (f, \alpha_1, \mathbf{0}, p_1)$, and, for all $i > 1$ let $r'_i = (p_{i-1}, \alpha_i, \mathbf{0}, p_i)$. Then $s r'_1 r'_2 r'_3 \dots$ is an accepting run of $\mathcal{A}$ on $w\alpha$, hence $w\alpha \in L_\omega(\mathcal{A})$. ◀

Observe that the construction in the proof of the lemma works the same way when we interpret $\mathcal{A}$ as a reachability-regular PA (every visit of an accepting state has the same good counter value; this argument is even true if we interpret $\mathcal{A}$ as a Büchi PA), showing the implication (1) $\Rightarrow$ (3).

▶ **Corollary 11.** *If $L \in \mathcal{L}^\omega_{\mathsf{PA},\mathsf{Reg}}$, then L is reachability-regular.*

For the backwards direction we need an auxiliary lemma, essentially stating that semi-linear sets over $C \subseteq (\mathbb{N} \cup \{\infty\})^d$ can be modified such that ∞-entries in vectors in C are replaced by arbitrary integers, and remain semi-linear.

▶ **Lemma 12.** *Let $C \subseteq (\mathbb{N} \cup \{\infty\})^d$ be semi-linear and $D \subseteq \{1, \dots, d\}$. Let $C_D \subseteq \mathbb{N}^d$ be the set obtained from C as follows.*

1. *Remove every vector $\mathbf{v} = (v_1, \dots, v_d)$ where $v_i = \infty$ for an $i \notin D$.*
2. *As long as C_D contains a vector $\mathbf{v} = (v_1, \dots, v_d)$ with $v_i = \infty$ for an $i \leq d$: replace $\mathbf{v}$ by all vectors of the form $(v_1, \dots v_{i-1}, z, v_{i+1}, \dots, v_d)$ for $z \in \mathbb{N}$.*

Then C_D is semi-linear.

Proof. For a vector $\mathbf{v} = (v_1, \dots, v_d) \in (\mathbb{N} \cup \{\infty\})^d$, let $\mathsf{Inf}(\mathbf{v}) = \{i \mid v_i = \infty\}$ denote the positions of ∞-entries in $\mathbf{v}$. Furthermore, let $\bar{\mathbf{v}} = (\bar{v}_1, \dots, \bar{v}_d)$ denote the vector obtained from $\mathbf{v}$ by replacing every ∞-entry by 0, i. e., $\bar{v}_i = 0$ if $v_i = \infty$, and $\bar{v}_i = v_i$ otherwise.

We carry out the following procedure for every linear set of the semi-linear set independently, hence we assume that $C = \{b_0 + b_1 z_1 + \dots + b_\ell z_\ell \mid z_1, \dots, z_\ell \in \mathbb{N}\}$ is linear. We also assume that there is no b_j with $\mathsf{Inf}(b_j) \not\subseteq D$, otherwise, we simply remove it.

Now, if $\mathsf{Inf}(b_0) \not\subseteq D$, then $C_D = \varnothing$, as this implies that every vector in C has an ∞-entry at an unwanted position (the first item of the lemma). Otherwise, $C_D = \{b_0 + \sum_{j \leq \ell} \bar{b}_j z_j + \sum_{i \in \mathsf{Inf}(b_j)} \mathbf{e}_i z_{ij} \mid z_j, z_{ij} \in \mathbb{N}\}$, which is linear by definition. ◀

We are now ready to prove the following lemma, showing the implication (2) ⇒ (1).

▶ **Lemma 13.** *If L is limit PA-recognizable, then $L \in \mathcal{L}^\omega_{\mathsf{PA},\mathsf{Reg}}$.*

Proof. Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ be an limit PA of dimension d. The idea is as follows. We guess a subset $D \subseteq \{1, \dots, d\}$ of counters whose values we expect to be ∞. Observe that every counter not in D has a finite value, hence for every such counter there is a point where all transitions do not increment the counter further. For every subset $D \subseteq \{1, \dots, d\}$ we decompose $\mathcal{A}$ into a PA and a TGBA. In the first step we construct a PA where every counter not in D reaches its final value and is verified. In the second step we construct a TGBA ensuring that for every counter in D at least one transition adding a non-zero value to that counter is used infinitely often. This can be encoded directly into the TGBA. Furthermore we delete all transitions that modify counters not in D.

Fix $D \subseteq \{1, \dots, d\}$ and $f \in F$, and define the PA $\mathcal{A}^D_f = (Q, \Sigma, q_0, \Delta, \{f\}, C_D)$ where C_D is defined as in Lemma 12. Furthermore, we define the TGBA $\mathcal{B}^D_f = (Q, \Sigma, f, \Delta^D, \mathcal{T}^D)$ where Δ^D contains the subset of transitions of Δ where the counters not in D have zero-values (just the transitions without vectors for the counters, as we construct a TGBA). On the other hand, for every counter i in D there is one acceptance component in $\mathcal{T}^D$ that contains exactly those transitions (again without vectors) where the ith counter has a non-zero value. Finally, we encode the condition that at least one accepting state in F needs to by seen infinitely often in $\mathcal{T}^D$ by further adding the component $\{(p, a, q) \in \Delta \mid q \in F\}$ (i. e. now we need to see an incoming transition of a state in F infinitely often).

We claim that $L_\omega(\mathcal{A}) = \bigcup_{D \subseteq \{1, \dots, d\}, f \in F} L(\mathcal{A}^D_f) \cdot L_\omega(\mathcal{B}^D_f)$, which by the comment below Theorem 9 and the equivalence of TGBA and Büchi automata implies the statement of the lemma. The details are presented in the appendix. ◀

The construction in Lemma 10 yields a limit PA whose semi-linear set C contains no vector with an ∞-entry. Hence, by this observation and the construction in the previous lemma we obtain the following corollary.

▶ **Corollary 14.** *For every limit PA there is an equivalent limit PA whose semi-linear set does not contain any ∞-entries.*

Finally we show the implication (3) ⇒ (1).

▶ **Lemma 15.** *If L is reachability-regular, then $L \in \mathcal{L}^\omega_{\mathsf{PA},\mathsf{Reg}}$.*

Proof. Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ be a reachability-regular PA. The intuition is as follows. a reachability-regular PA just needs to verify the counters a single time. Hence, we can recognize the prefixes of infinite words $\alpha \in B_\omega(\mathcal{A})$ that generate the accepting hit with a PA. Further checking that an accepting state is seen infinitely often can be done with a Büchi automaton.

Fix $f \in F$ and let $\mathcal{A}_f = (Q, \Sigma, q_0, \Delta, \{f\}, C)$ be the PA that is, syntactically equal to $\mathcal{A}$ with the only difference that f is the only accepting state. Similarly, let $\mathcal{B}_f = (Q, \Sigma, f, \{(p, a, q) \mid (p, a, \mathbf{v}, q) \in \Delta\}, F)$ be the Büchi automaton obtained from $\mathcal{A}$ by setting f as the initial state and the forgetting the vector labels.

We claim that $RR_\omega(\mathcal{A}) = \bigcup_{f \in F} L(\mathcal{A}_f) \cdot L_\omega(\mathcal{B}_f)$.

$\Rightarrow$ To show $RR_\omega(\mathcal{A}) \subseteq \bigcup_{f \in F} L(\mathcal{A}_f) \cdot L_\omega(\mathcal{B}_f)$, let $\alpha \in B_\omega(\mathcal{A})$ with accepting run $r = r_1 r_2 r_3 \dots$ where $r_i = (p_{i-1}, \alpha_i, \mathbf{v}_i, p_i)$. Let k be arbitrary such that there is an accepting hit in r_k (such a k exists by definition) and consider the prefix $\alpha[1, k]$. Obviously $r_1 \dots r_k$ is an accepting run of $\mathcal{A}_{p_k}$ on $\alpha[1, k]$. Furthermore, there are infinitely many j such that $p_j \in F$ by definition. In particular, there are also infinitely many $j \geq k$ with this property. Let $r'_i = (p_{i-1}, \alpha_i, p_i)$ for all $i > k$. Then $r'_{k+1} r'_{k+2} \dots$ is an accepting run of $\mathcal{B}_{p_k}$ on $\alpha_{k+1} \alpha_{k+2} \dots$ (recall that p_k is the initial state of $\mathcal{B}_{p_k}$). Hence we have $\alpha[1, k] \in L(\mathcal{A}_{p_k})$ and $\alpha_{k+1} \alpha_{k+2} \dots \in L_\omega(\mathcal{B}_{p_k})$.

$\Leftarrow$ To show $\bigcup_{f \in F} L(\mathcal{A}_f) \cdot L_\omega(\mathcal{B}_f) \subseteq RR_\omega(\mathcal{A})$, let $w \in L(\mathcal{A}_f)$ and $\beta \in L_\omega(\mathcal{B}_f)$ for some $f \in F$. We show $w\beta \in B_\omega(\mathcal{A})$. Let $s = s_1 \dots s_n$ be an accepting run of $\mathcal{A}_f$ on w, which ends in the accepting state f with $\rho(s) \in C$ by definition. Furthermore, let $r = r_1 r_2 r_3 \dots$ be an accepting run of $\mathcal{B}_f^D$ on β which starts in the accepting state f by definition. It is now easily verified that sr' with $r' = r'_1 r'_2 r'_3 \dots$ where $r'_i = (p_{i-1}, \alpha_i, \mathbf{v}_i, p_i)$ (for an arbitrary $\mathbf{v}_i$ such that $r'_i \in \Delta$) is an accepting run of $\mathcal{A}$ on $w\beta$, as there is an accepting hit in s_n, and the (infinitely many) visits of an accepting state in r translate one-to-one, hence $w\beta \in B_\omega(\mathcal{A})$. ◀

As shown in Lemma 7, the class of Büchi PA-recognizable ω-languages is equivalent to the class of ω-languages of the form $\bigcup_i U_i V_i^\omega$ where U_i and V_i are Parikh-recognizable, but the PA for V_i is restricted in such a way that the initial state is the only accepting state and the set is a homogeneous linear set. Observe that for every regular language L there is a Büchi automaton $\mathcal{A}$ where the initial state is the only accepting state with $L_\omega(\mathcal{A}) = L^\omega$ (see e.g. [29, Lemma 1.2]). Hence, $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{Reg}}$ is a subset of the class of Büchi PA-recognizable ω-languages. This inclusion is also strict, as witnessed by the Büchi PA in Example 1 which has the mentioned property.

▶ **Corollary 16.** *The class $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{Reg}}$ is a strict subclass of the class of Büchi PA-recognizable ω-languages.*

We finish this subsection by observing that (complete) reachability PA capture a subclass of $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{Reg}}$ where, due to completeness, all $V_i = \Sigma$.

▶ **Observation 17.** *The following are equivalent for all ω-languages $L \subseteq \Sigma^\omega$.*

1. *L is of the form $\bigcup_i U_i \Sigma^\omega$ where $U_i \subseteq \Sigma^*$ is Parikh-recognizable.*
2. *L is reachability PA-recognizable.*

4.3 Characterization of $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{PA}}$ and $\mathcal{L}^\omega_{\mathsf{Reg},\mathsf{PA}}$

In this section we give a characterization of $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{PA}}$ and a characterization of $\mathcal{L}^\omega_{\mathsf{Reg},\mathsf{PA}}$. As mentioned in the beginning of this section, reset PA are too strong to capture this class. However, restrictions of strong reset PA are good candidates to capture $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{PA}}$ as well as $\mathcal{L}^{\omega}_{\mathsf{Reg,PA}}$. In fact we show that it is sufficient to restrict the appearances of accepting states to capture $\mathcal{L}^{\omega}_{\mathsf{PA,PA}}$, as specified by the first theorem of this subsection. Further restricting the vectors yields a model capturing $\mathcal{L}^{\omega}_{\mathsf{Reg,PA}}$, as specified in the second theorem of this subsection. Recall that the condensation of $\mathcal{A}$ is the DAG of strong components of the underlying graph of $\mathcal{A}$.

▶ **Theorem 18.** *The following are equivalent for all ω-languages $L \subseteq \Sigma^\omega$.*

1. *L is of the form $\bigcup_i U_i V_i^\omega$, where $U_i, V_i \subseteq \Sigma^*$ are Parikh-recognizable.*
2. *L is recognized by a strong reset PA $\mathcal{A}$ with the property that accepting states appear only in the leaves of the condensation of $\mathcal{A}$, and there is at most one accepting state per leaf.*

Proof. (1) $\Rightarrow$ (2). Let $\mathcal{A}_i = (Q_i, \Sigma, q_i, \Delta_i, F_i)$ for $i \in \{1, 2\}$ be PA and let $L = L(\mathcal{A}_1) \cdot L(\mathcal{A}_2)^\omega$. By Lemma 4 we may assume that $\mathcal{A}_2$ is normalized (recall that by Corollary 5 this implies $SR_\omega(\mathcal{A}_2) = L(\mathcal{A}_2)^\omega$) and hence write $L = L(\mathcal{A}_1) \cdot SR_\omega(\mathcal{A}_2)$. As pointed out in the proof of Lemma 3, we can construct a reset PA $\mathcal{A}$ that recognizes L such that only the accepting states of $\mathcal{A}_2$ remain accepting in $\mathcal{A}$. As $\mathcal{A}_2$ is normalized, this means that only q_2 is accepting in $\mathcal{A}$. Hence $\mathcal{A}$ satisfies the property of the theorem. Finally observe that the construction in Lemma 2 maintains this property, implying that the construction presented in Lemma 6 always yields a reset PA of the desired form. ◀

(2) $\Rightarrow$ (1). Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ be a strong reset PA of dimension d with the property of the theorem. Let $f \in F$ and let $\mathcal{A}_f = (Q, \Sigma, q_0, \Delta_f, \{f\}, C \cdot \{1\})$ with $\Delta_f = \{p, a, \mathbf{v} \cdot 0, q) \mid (p, a, \mathbf{v}, q) \in \Delta, q \neq f\} \cup \{(p, a, \mathbf{v} \cdot 1, f) \mid (p, a, \mathbf{v}, f) \in \Delta\}$ be the PA of dimension $d + 1$ obtained from $\mathcal{A}$ by setting f as the only accepting state with an additional counter that is 0 at every transition except the incoming transitions of f, where the counter is set to 1. Additionally all vectors in C are concatenated with 1. Similarly, let $\mathcal{A}_{f,f} = (Q, \Sigma, f, \Delta_f, \{f\}, C \cdot \{1\})$ be the PA of dimension $d + 1$ obtained from $\mathcal{A}_f$ by setting f as the initial state.

$\Rightarrow$ To show $SR_\omega(\mathcal{A}) \subseteq \bigcup_{f \in F} L(\mathcal{A}_f) \cdot L(\mathcal{A}_{f,f})^\omega$, let $\alpha \in S_\omega(\mathcal{A})$ with accepting run $r = r_1 r_2 r_3 \dots$ where $r_i = (p_{i-1}, \alpha_i, \mathbf{v}_i, p_i)$. Let $k_1 < k_2 < \dots$ be the positions of accepting states in r, i. e., $p_{k_i} \in F$ for all $i \geq 1$. First observe that the property in the theorem implies $p_{k_i} = p_{k_j}$ for all $i, j \geq 1$, i. e., no two distinct accepting states appear in r, since accepting states appear only in different leaves of the condensation of $\mathcal{A}$.

For $j \geq 1$ define $r'_j = (p_{j-1}, \alpha_j, \mathbf{v}_j \cdot 0, p_j)$ if $j \neq k_i$ for all $i \geq 1$, and $r'_j = (p_{j-1}, \alpha_j, \mathbf{v}_j \cdot 1, p_j)$ if $j = k_i$ for some $i \geq 1$, i. e., we replace every transition r_j by the corresponding transition in Δ_f.

Now consider the partial run $r_1 \dots r_{k_1}$ and observe that $p_i \neq p_{k_1}$ for all $i < k_1$, and $\rho(r_1 \dots r_{k_1}) \in C$ by the definition of strong reset PA. Hence $r' = r'_1 \dots r'_{k_1}$ is an accepting run of $\mathcal{A}_{p_{k_1}}$ on $\alpha[1, k_1]$, as only a single accepting state appears in r', the newly introduced counter has a value of 1 when entering p_{k_1}, i. e., $\rho(r') \in C \cdot \{1\}$, hence $\alpha[1, k_1] \in L(\mathcal{A}_{p_{k_1}})$.

Finally, we show that $\alpha[k_i + 1, k_{i+1}] \in L(\mathcal{A}_{p_{k_1}, p_{k_1}})$. Observe that $r'_{k_i+1} \dots r'_{k_{i+1}}$ is an accepting run of $\mathcal{A}_{p_{k_1}, p_{k_1}}$ on $\alpha[k_i + 1, k_{i+1}]$: we have $\rho(r_{k_i+1} \dots r_{k_{i+1}}) = \mathbf{v} \in C$ by definition. Again, as only a single accepting state appears in $r'_{k_i+1} \dots r'_{k_{i+1}}$, we have $\rho(r'_{k_i+1} \dots r'_{k_{i+1}}) = \mathbf{v} \cdot 1 \in C \cdot \{1\}$, and hence $\alpha[k_i+1, k_{i+1}] \in L(\mathcal{A}_{p_{k_1}, p_{k_1}})$. We conclude $\alpha \in L(\mathcal{A}_{p_{k_1}}) \cdot L(\mathcal{A}_{p_{k_1}, p_{k_1}})^\omega$.

$\Leftarrow$ To show $\bigcup_{f \in F} L(\mathcal{A}_f) \cdot L(\mathcal{A}_{f,f})^\omega \subseteq SR_\omega(\mathcal{A})$, let $u \in L(\mathcal{A}_f)$, and $v_1, v_2, \dots \in L(\mathcal{A}_{f,f})$ for some $f \in F$. We show that $uv_1v_2 \dots \in SR_\omega(\mathcal{A})$.

First let $u = u_1 \dots u_n$ and $r' = r'_1 \dots r'_n$ with $r'_i = (p_{i-1}, u_i, \mathbf{v}_i \cdot c_i, p_i)$, where $c_i \in \{0,1\}$, be an accepting run of $\mathcal{A}_f$ on u. Observe that $\rho(r') \in C \cdot \{1\}$, hence $\sum_{i \leq n} c_i = 1$, i. e., p_n is the only occurrence of an accepting state in r' (if there was another, say p_j, then $c_j = 1$ by the choice of Δ_f, hence $\sum_{i \leq n} c_i > 1$, a contradiction). For all $1 \leq i \leq n$ let $r_i = (p_{i-1}, u_i, \mathbf{v}_i, p_i)$. Then $r_1 \dots r_n$ is a partial run of $\mathcal{A}$ on w with $\rho(r_1 \dots r_n) \in C$ and $p_n = f$.

Similarly, no run of $\mathcal{A}_{f,f}$ on any v_i visits an accepting state before reading the last symbol, hence we continue the run from r_n on $v_1, v_2, \dots$ using the same argument. Hence $uv_1v_2 \dots \in SR_\omega(\mathcal{A})$, concluding the proof. ◀

As a side product of the proof of Theorem 18 we get the following corollary, which is in general not true for arbitrary reset PA.

▶ **Corollary 19.** *Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ be a strong reset PA with the property that accepting states appear only in the leaves of the condensation of $\mathcal{A}$, and there is at most one accepting state per leaf. Then we have $SR_\omega(\mathcal{A}) = \bigcup_{f \in F} S_\omega(Q, \Sigma, q_0, \Delta, \{f\}, C)$.*

By even further restricting the power of strong reset PA, we get the following characterization of $\mathcal{L}^\omega_{\mathsf{Reg,PA}}$.

▶ **Theorem 20** ($\star$)**.** *The following are equivalent for all ω-languages $L \subseteq \Sigma^\omega$.*

1. *L is of the form $\bigcup_i U_i V_i^\omega$, where $U_i \subseteq \Sigma^*$ is regular and $V_i \subseteq \Sigma^*$ is Parikh-recognizable.*
2. *L is recognized by a strong reset PA $\mathcal{A}$ with the following properties.*
   - **(a)** *At most one state q per leaf of the condensation of $\mathcal{A}$ may have incoming transitions from outside the leaf, this state q is the only accepting state in the leaf, and there are no accepting states in non-leaves.*
   - **(b)** *only transitions connecting states in a leaf may be labeled with a non-zero vector.*

Observe that property (a) is a stronger property than the one of Theorem 18, hence, strong reset PA with this restriction are at most as powerful as those that characterize $\mathcal{L}^\omega_{\mathsf{PA,PA}}$. However, as a side product of the proof we get that property (a) is equivalent to the property of Theorem 18. Hence, property (b) is mandatory to sufficiently weaken strong reset PA such that they capture $\mathcal{L}^\omega_{\mathsf{Reg,PA}}$. In fact, using the notion of normalization, we can re-use most of the ideas in the proof of Theorem 18.

5 Blind counter machines and ε-elimination

As mentioned in the introduction, blind counter machines as an extension of automata with counting mechanisms were already introduced and studied in the 70s [18]. Over finite words they are equivalent to Parikh automata [26]. Blind counter machines over infinite words were first considered by Fernau and Stiebe [14]. In this section we first recall the definition of blind counter machines as introduced by Fernau and Stiebe [14]. The definition of these automata admits ε-transitions. It is easily observed that Büchi PA with ε-transitions are equivalent to blind counter machines. Therefore, we extend all Parikh automata models studied in this paper with ε-transitions and consider the natural question whether they admit ε-elimination (over infinite words). We show that almost all models allow ε-elimination, the exception being safety and co-Büchi PA. For the latter two models we observe that ε-transitions allow to encode ω-regular conditions, meaning that such transitions give the models enough power such that they can recognize all ω-regular languages.

A *blind k-counter machine* (CM) is a quintuple $\mathcal{M} = (Q, \Sigma, q_0, \Delta, F)$ where Q, Σ, q_0 and F are defined as for NFA, and $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \mathbb{Z}^k \times Q$ is a finite set of *integer labeled transitions*. In particular, the transitions of Δ are labeled with possibly negative integer vectors. Observe that ε-transitions are allowed.

A *configuration* for an infinite word $\alpha = \alpha_1\alpha_2\alpha_3\dots$ of $\mathcal{M}$ is a tuple of the form $c = (p, \alpha_1\dots\alpha_i, \alpha_{i+1}\alpha_{i+2}\dots, \mathbf{v}) \in Q \times \Sigma^* \times \Sigma^\omega \times \mathbb{Z}^k$ for some $i \geq 0$. A configuration c *derives* into a configuration c', written $c \vdash c'$, if either $c' = (q, \alpha_1\dots\alpha_{i+1}, \alpha_{i+2}\dots, \mathbf{v}+\mathbf{u})$ and $(p, \alpha_{i+1}, \mathbf{u}, q) \in \Delta$, or $c' = (q, \alpha_1\dots\alpha_i, \alpha_{i+1}\alpha_{i+2}\dots, \mathbf{v}+\mathbf{u})$ and $(p, \varepsilon, \mathbf{u}, q) \in \Delta$. $\mathcal{M}$ *accepts* an infinite word α if there is an infinite sequence of configuration derivations $c_1 \vdash c_2 \vdash c_3 \vdash \dots$ with $c_1 = (q_0, \varepsilon, \alpha, \mathbf{0})$ such that for infinitely many i we have $c_i = (p_i, \alpha_1\dots\alpha_j, \alpha_{j+1}\alpha_{j+2}\dots, \mathbf{0})$ with $p_i \in F$ and for all $j \geq 1$ there is a configuration of the form $(p, \alpha_1\dots\alpha_j, \alpha_{j+1}\alpha_{j+2}\dots, \mathbf{v})$ for some $p \in Q$ and $\mathbf{v} \in \mathbb{Z}^k$ in the sequence. That is, a word is accepted if we infinitely often visit an accepting state when the counters are $\mathbf{0}$, and every symbol of α is read at some point. We define the ω-language recognized by $\mathcal{M}$ as $L_\omega(\mathcal{M}) = \{\alpha \in \Sigma^\omega \mid \mathcal{M} \text{ accepts } \alpha\}$.

Parikh automata naturally generalize to Parikh automata with ε-transitions. An ε-PA is a tuple $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{E}, F, C)$ where $\mathcal{E} \subseteq Q \times \{\varepsilon\} \times \mathbb{N}^d \times Q$ is a finite set of *labeled ε-transitions*, and all other entries are defined as for PA. A run of $\mathcal{A}$ on an infinite word $\alpha_1\alpha_2\alpha_3\dots$ is an infinite sequence of transitions $r \in (\mathcal{E}^*\Delta)^\omega$, say $r = r_1r_2r_3\dots$ with $r_i = (p_{i-1}, \gamma_i, \mathbf{v}_i, p_i)$ such that $p_0 = q_0$, and $\gamma_i = \varepsilon$ if $r_i \in \mathcal{E}$, and $\gamma_i = \alpha_j$ if $r_i \in \Delta$ is the j-th occurrence of a (non-ε) transition in r. The acceptance conditions of the models translate to runs of ε-PA in the obvious way. We use terms like ε-safety PA, ε-reachability PA, etc, to denote an ε-PA with the respective acceptance condition.

Note that we can treat every PA as an ε-PA, that is, a PA $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ is equivalent to the ε-PA $\mathcal{A}' = (Q, \Sigma, q_0, \Delta, \varnothing, F, C)$.

5.1 Equivalence of blind counter machines with Büchi PA

We start with the following simple observation.

▶ **Lemma 21** ($\star$). *CM and ε-Büchi PA are equivalent.*

5.2 ε-elimination for Parikh automata

We now show that almost all PA models admit ε-elimination. We first consider Büchi PA, where ε-elimination implies the equivalence of blind counter machines and Büchi PA by Lemma 21. We provided a direct but quite complicated proof in the manuscript [19]. We thank Georg Zetzsche for outlining a much simpler proof, which we present here.

▶ **Theorem 22.** *ε-Büchi PA admit ε-elimination.*

Proof. Observe that the construction in Lemma 21 translates ε-free CM into ε-free Büchi PA. We can hence translate a given Büchi PA into a CM and eliminate ε-transitions and then translate back into a Büchi PA. Therefore, all we need to show is that CM admit ε-elimination.

To show that CM admit ε-elimination we observe that

$$L \text{ is recognized by a CM} \quad \iff \quad L = \bigcup_i U_i V_i^\omega,$$

where U_i is a language of finite words that is recognized by a CM and V_i is a language of finite words that is recognized by a CM where $F = \{q_0\}$. The proof of this observation is very similar to the proof of Lemma 7 and we leave the details to the reader.

As shown in [18, 27, 30], CM on finite words admit ε-elimination. Furthermore, the proof technique established in [30, Lemma 7.7] it is immediate that the condition that $F = \{q_0\}$ is preserved. We obtain ε-free CM $\mathcal{A}_i'$ and $\mathcal{B}_i'$ for the languages U_i and V_i. Using the construction of [26], we can translate $\mathcal{A}_i'$ and $\mathcal{B}_i'$ into PA $\mathcal{A}_i$ and $\mathcal{B}_i$, where the $\mathcal{B}_i$ satisfy $F_i = \{q_0\}$ and the sets C_i are homogeneous linear sets (Theorem 32 of [26]). Now the statement follows by Lemma 7. ◀

We continue with ε-reachability, ε-reachability-regular and ε-limit PA, as we show ε-elimination using the same technique for these models. As shown in Observation 17 and Theorem 9, the class of ω-languages recognized by reachability PA coincides with the class of ω-languages of the form $\bigcup_i U_i\Sigma^\omega$ for Parikh-recognizable U_i, and the class of reachability-regular and limit PA-recognizable ω-languages coincides with the class of ω-languages of the form $\bigcup_i U_i V_i^\omega$ for Parikh-recognizable U_i and regular V_i, respectively. It is well-known that NFA and PA on finite words are closed under homomorphisms and hence admit ε-elimination [26] (as a consequence of [27, Proposition II.11], ε-transitions can even be eliminated without changing the semi-linear set). The characterizations allow us to reduce ε-elimination of these infinite word PA to the finite case.

▶ **Lemma 23** ($\star$). *ε-reachability, ε-reachability-regular, and ε-limit PA admit ε-elimination.*

Finally we show that safety and co-Büchi PA do not admit ε-elimination.

▶ **Lemma 24.** *ε-safety PA and ε-co-Büchi PA do not admit ε-elimination.*

Proof. Consider the automaton $\mathcal{A}$ in Figure 3 with $C = \{(z, z') \mid z' \geq z\}$.

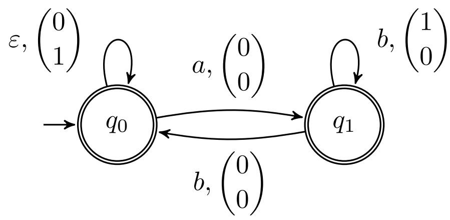


Figure 3 The ε-PA with $C = \{(z, z') \mid z' \geq z\}$ for the proof of Lemma 24.

If we interpret $\mathcal{A}$ as an ε-safety or ε-co-Büchi PA, we have we have $S_\omega(\mathcal{A}) = CB_\omega(\mathcal{A}) = \{ab^+\}^\omega$. This ω-language is neither safety PA nor co-Büchi PA-recognizable (one can easily adapt the proof in [22] showing that $\{\alpha \in \{a,b\}^\omega \mid |\alpha|_a = \infty\}$ is neither safety PA nor co-Büchi PA-recognizable).

Observe how $\mathcal{A}$ utilizes the ε-transition to enforce that q_0 is seen infinitely often: whenever the b-loop on q_1 is used, the first counter increments. The semi-linear set states that at no point the first counter value may be greater than the second counter value which can only be increased using the ε-loop on q_0. Hence, any infinite word accepted by $\mathcal{A}$ may contain arbitrary infixes of the form b^n for $n < \infty$, as the automaton can use the ε-loop on q_0 at least n times before, but not b^ω. ◀

As a consequence of the previous proof we show that ε-safety PA and ε-co-Büchi PA recognize all ω-regular languages, as the presented trick can be used to encode ω-regular conditions, that is ε-transitions can be used to enforce that at least one state of a subset of states needs to be visited infinitely often.

▶ **Lemma 25** ($\star$). *Every ω-regular language is ε-safety PA and ε-co-Büchi recognizable.*

Finally we show that strong ε-reset PA and weak ε-reset PA admit ε-elimination. We show that these two models are equivalent. Hence to show this statement we only need to argue that strong ε-reset PA admit ε-elimination.

▶ **Lemma 26** ($\star$). *Every strong ε-reset PA $\mathcal{A}$ is equivalent to a weak ε-reset PA $\mathcal{A}'$ that has the same set of states and uses one additional counter. If $\mathcal{A}$ is a strong reset PA, then $\mathcal{A}'$ is a weak reset PA.*

▶ **Lemma 27** ($\star$). *Every weak ε-reset PA $\mathcal{A}$ is equivalent to a strong ε-reset PA $\mathcal{A}'$ with at most twice the number of states and the same number of counters. If $\mathcal{A}$ is a strong reset PA, then $\mathcal{A}'$ is a weak reset PA.*

▶ **Lemma 28** ($\star$). *Strong ε-reset PA admit ε-elimination.*

6 Decision problems

As shown by Guha et al. [22], the results for common decision problems translate from the finite case to reachability PA and Büchi PA, that is, non-emptiness is NP-complete, and universality (and hence inclusion and equivalence) are undecidable. We show that these results translate to reset PA (which are more expressive), even if we allow ε-transitions (which does not increase their expressiveness but our ε-elimination procedure constructs an equivalent reset PA of super-polynomial size). Hence, (ε-)reset PA are a powerful model that can still be used for algorithmic applications, such as the model checking problem.

The main reason for this is that the ω-languages recognized by reset PA are ultimately periodic, meaning that whenever a reset PA accepts at least one infinite word, then it also accepts an infinite word of the form uv^ω.

▶ **Lemma 29** ($\star$). *Let $\mathcal{A}$ be an ε-reset PA. If $SR_\omega(\mathcal{A}) \neq \varnothing$, then $\mathcal{A}$ accepts an infinite word of the form uv^ω.*

As a consequence, we can reduce non-emptiness for reset PA to the finite word case, as clarified in the following lemma.

▶ **Lemma 30** ($\star$). *Non-emptiness for ε-reset PA is* NP*-complete.*

Furthermore, we study the following membership problem for automata processing infinite words. Given an automaton $\mathcal{A}$ and finite words u, v, does $\mathcal{A}$ accept uv^ω?

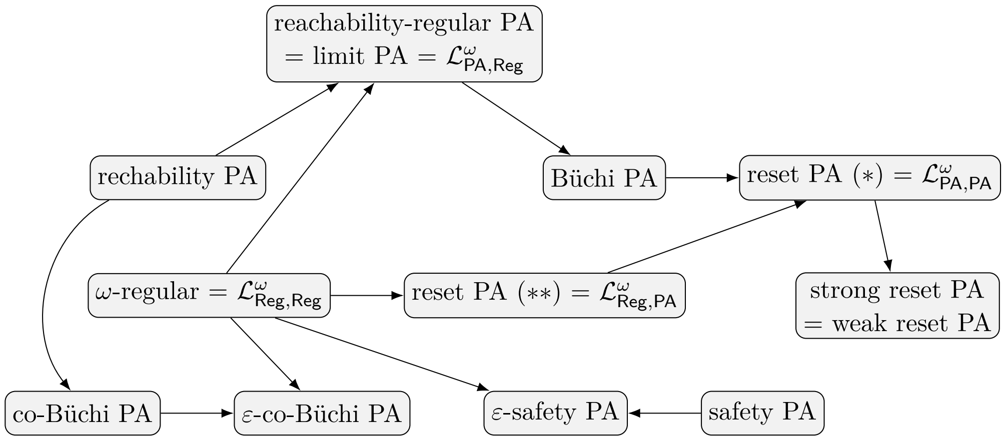
Note that we can always construct a safety automaton that recognizes uv^ω and no other infinite word with $|uv|$ many states. Recall that every state of a safety automaton is accepting. We show that the intersection of a reset PA-recognizable ω-language and a safety automaton-recognizable ω-language remains reset PA-recognizable using a product construction which is computable in polynomial time. Hence, we can reduce the membership problem to the non-emptiness the standard way.

▶ **Lemma 31** ($\star$). *The class of reset PA-recognizable ω-languages is closed under intersection with safety automata-recognizable ω-languages.*

As the membership problem for PA (on finite words) is NP-complete [15], and the construction in the previous lemma can be computed efficiently, we obtain the following result.

▶ **Corollary 32.** *Membership for ε-reset PA is* NP*-complete.*

Finally, we observe that universality, inclusion and equivalence remain undecidable for (ε-)reset PA, as these problems are already undecidable for Büchi PA [22] and the constructions showing that the class of Büchi PA-recognizable ω-languages is a subclass of $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{PA}}$, and that $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{PA}}$ is a subclass of the class of reset PA-recognizable ω-languages are effective.



(∗) At most one state q per leaf of $C(\mathcal{A})$ may have incoming transitions from outside the leaf, this state q is the only accepting state in the leaf, and there are no accepting states in non-leaves;
(∗∗) and only transitions connecting states in leaves may be labeled with non-zero vectors.

Figure 4 Overview of our results. Arrows mean strict inclusions. If not explicitly shown otherwise, all models are equivalent to their ε-counterparts.

7 Conclusion

We conclude by giving an overview of all characterizations and inclusions shown in this paper, as depicted in Figure 4.

Recall the ω-languages motivated by the model checking problem from the introduction, namely $\{\alpha \in \{a,b,c\}^\omega \mid$ there are infinitely many prefixes w of α with $|w|_a > |w|_b + |w|_c\}$, representing unfair resource distributions of an operating system, and $\{\alpha \in \{p,c\}^\omega \mid$ there is a prefix w of α with $|w|_c > |w|_p\}$, representing invalid computations in a producer-consumer setting. Both of these ω-languages are Reset PA-recognizable (in fact, the first is Büchi PA-recognizable and the second is even reachability PA-recognizable). As mentioned, in a common approach we are given a system represented as a Kripke structure K, and a specification of counter-examples given as an automaton, e.g. a reset PA $\mathcal{A}$. By moving the labels of the states of K to its transitions, we can see a Kripke structure as a safety automaton $\mathcal{A}_K$ (see [10, Theorem 28] for details). As every state of a safety automaton is accepting, we can easily find a reset automaton recognizing all bad computations of K (that is the intersection of the ω-languages recognized by $\mathcal{A}_K$ and $\mathcal{A}$) by Lemma 31. As (non-)emptiness is decidable for reset PA, we can solve the model-checking problem by computing the product automaton of $\mathcal{A}_K$ and $\mathcal{A}$ and testing for emptiness, which is in coNP by Lemma 30.

References

1 Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, 2008.

2 Brenda S. Baker and Ronald V. Book. Reversal-bounded multipushdown machines. *Journal of Computer and System Sciences*, 8(3):315–332, 1974.

3 Pascal Baumann, Flavio D'Alessandro, Moses Ganardi, Oscar Ibarra, Ian McQuillan, Lia Schütze, and Georg Zetzsche. Unboundedness problems for machines with reversal-bounded counters. In *Foundations of Software Science and Computation Structures*, pages 240–264, Cham, 2023. Springer Nature Switzerland.

4 J. Richard Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92, 1960.

5 Michaël Cadilhac. *Automates à contraintes semilinéaires= Automata with a semilinear constraint*. PhD thesis, University of Montréal, 2013.

6 Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. On the expressiveness of Parikh automata and related models. In *Third Workshop on Non-Classical Models for Automata and Applications - NCMA 2011*, volume 282 of *books@ocg.at*, pages 103–119. Austrian Computer Society, 2011.

7 Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. Affine Parikh automata. *RAIRO Theor. Informatics Appl.*, 46(4):511–545, 2012.

8 Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. Bounded Parikh automata. *Int. J. Found. Comput. Sci.*, 23(8):1691–1710, 2012.

9 Edmund M Clarke, Orna Grumberg, and Doron A. Peled. *Model checking*. The MIT Press, London, Cambridge, 1999.

10 Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem. *Handbook of Model Checking*. Springer Publishing Company, Incorporated, 1st edition, 2018.

11 Luc Dartois, Emmanuel Filiot, and Jean-Marc Talbot. Two-way Parikh automata with a visibly pushdown stack. In *Foundations of Software Science and Computation Structures - 22nd International Conference, FOSSACS 2019*, volume 11425 of *Lecture Notes in Computer Science*, pages 189–206. Springer, 2019.

12 Enzo Erlich, Shibashis Guha, Ismaël Jecker, Karoliina Lehtinen, and Martin Zimmermann. History-deterministic Parikh automata. *arXiv preprint*, 2022. `arXiv:2209.07745`.

13 Henning Fernau and Ralf Stiebe. Sequential grammars and automata with valences. *Theoretical Computer Science*, 276(1):377–405, 2002.

14 Henning Fernau and Ralf Stiebe. Blind counter automata on omega-words. *Fundam. Inform.*, 83:51–64, 2008.

15 Diego Figueira and Leonid Libkin. Path logics for querying graphs: Combining expressiveness and efficiency. In *Proceedings of the 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, LICS '15, pages 329–340. IEEE, 2015.

16 Emmanuel Filiot, Shibashis Guha, and Nicolas Mazzocchi. Two-way Parikh automata. In *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019*, volume 150 of *LIPIcs*, pages 40:1–40:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.

17 Dimitra Giannakopoulou and Flavio Lerda. From states to transitions: Improving translation of LTL formulae to Büchi automata. In *Formal Techniques for (Networked and) Distributed Systems*, 2002.

18 Sheila A. Greibach. Remarks on blind and partially blind one-way multicounter machines. *Theoretical Computer Science*, 7(3):311–324, 1978.

19 Mario Grobler, Leif Sabellek, and Sebastian Siebertz. Parikh automata on infinite words. *arXiv preprint*, 2023. `arXiv:2301.08969`.

20 Mario Grobler, Leif Sabellek, and Sebastian Siebertz. Remarks on Parikh-recognizable omega-languages. *arXiv preprint*, 2023. `arXiv:2307.07238`.

21 Mario Grobler and Sebastian Siebertz. Büchi-like characterizations for Parikh-recognizable omega-languages. *arXiv preprint*, 2023. `arXiv:2302.04087`.

22 Shibashis Guha, Ismaël Jecker, Karoliina Lehtinen, and Martin Zimmermann. Parikh Automata over Infinite Words. In *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2022)*, volume 250 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 40:1–40:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

23 Hendrik Jan Hoogeboom. Context-free valence grammars - revisited. In *Developments in Language Theory*, pages 293–303, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

24 Oscar H. Ibarra. Reversal-bounded multicounter machines and their decision problems. *J. ACM*, 25(1):116–133, 1978.

25 Wong Karianto. Parikh automata with pushdown stack. *Diplomarbeit, RWTH Aachen*, 2004.

26 Felix Klaedtke and Harald Rueß. Monadic second-order logics with cardinalities. In *Automata, Languages and Programming*, pages 681–696, Berlin, Heidelberg, 2003. Springer.

27 Michel Latteux. Cônes rationnels commutatifs. *Journal of Computer and System Sciences*, 18(3):307–333, 1979.

28 Victor Mitrana and Ralf Stiebe. Extended finite automata over groups. *Discrete Applied Mathematics*, 108(3):287–300, 2001.

29 Wolfgang Thomas. *Automata on Infinite Objects*, pages 133–191. MIT Press, Cambridge, MA, USA, 1991.

30 Georg Zetzsche. Silent transitions in automata with storage. In *Automata, Languages, and Programming*, pages 434–445, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.