Büchi-like characterizations for Parikh-recognizable omega-languages

Mario Grobler

University of Bremen, Bremen, Germany

Sebastian Siebertz

University of Bremen, Bremen, Germany

Abstract

Büchi's theorem states that ω -regular languages are characterized as languages of the form $\bigcup_i U_i V_i^{\omega}$, where U_i and V_i are regular languages. Parikh automata are automata on finite words whose transitions are equipped with vectors of positive integers, whose sum can be tested for membership in a given semi-linear set. We give an intuitive automata theoretic characterization of languages of the form $U_i V_i^{\omega}$, where U_i and V_i are Parikh-recognizable. Furthermore, we show that the class of such languages, where U_i is Parikh-recognizable and V_i is regular is exactly captured by a model proposed by Klaedtke and Ruess [Automata, Languages and Programming, 2003], which again is equivalent to (a small modification of) reachability Parikh automata introduced by Guha et al. [FSTTCS, 2022]. We finish this study by introducing a model that captures exactly such languages for regular U_i and Parikh-recognizable V_i .

Keywords: Automata theory, Parikh automata, infinite words, Büchi's theorem

1. Introduction

In his groundbreaking work [4] from 1960 Büchi initiated the study of ω -regular languages and introduced Büchi automata. By his famous theorem 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. One shortcoming of Büchi automata, and also of many more powerful models, is that they cannot count. For example, the language $\{a^nb^nc^n \mid n \in \mathbb{N}\}^{\omega}$ is not ω -regular, and not even ω -context-free. This shortcoming led to the study of automata on infinite words with counters, see e.g. [1, 3, 9].

Parikh automata (PA) are another model of automata (on finite words) with counters [8]. A PA with d counters is a non-deterministic finite automaton that is additionally equipped with a semi-linear set C. Furthermore, every transition is equipped with a d-tuple of non-negative integers and every time a transition is used, the counters are

Email addresses: grobler@uni-bremen.de (Mario Grobler), siebertz@uni-bremen.de (Sebastian Siebertz)

incremented by the values in the tuple accordingly. A finite input word is accepted if the PA ends in a final state and additionally, the resulting d-tuple in the counters lies in C. The class of languages recognized by PA contains all regular languages, and even some, but not all, context-sensitive languages, e.g. the language $\{a^nb^nc^n \mid n \in \mathbb{N}\}$.

Recently, several possible extensions of Parikh automata on infinite words were proposed and studied by Grobler et al. [6] and Guha et al. [7]. In fact, it turns out that one of the models proposed in [6] is equivalent to synchronous blind counter machines, which were introduced by Fernau and Stiebe [5]. Fernau and Stiebe also considered the class of all ω -languages of the form $\bigcup_i U_i V_i^{\omega}$, where the U_i, V_i are Parikh-recognizable languages of finite words. They called this class \mathcal{K}_* and proved that the class of ω -languages recognized by blind counter machines is a proper subset of \mathcal{K}_* .

In the light of Büchi's famous theorem it is a natural question to find an automata theoretic characterization of the class \mathcal{K}_* . In fact, more generally, 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. In this work we provide automata theoretic characterizations of the other three classes.

We first introduce the new model of *limit Parikh Büchi automata* (LPBA), which was suggested in the concluding remarks of Klaedtke and Ruess [8]. An LPBA accepts an infinite word if an accepting state is visited infinitely often (satisfies the Büchi condition) and the infinite sum of the counters belongs to the semi-linear set, which for this purpose is extended with the symbol ∞ if the sum of some counter diverges (satisfies the newly introduced *limit Parikh condition*).

We also introduce a new model, which is obtained by a small modification of reachability Parikh automata as introduced by Guha et al. [7], that we call reachability Parikh Büchi automata (RPBA). An RPBA accepts an infinite word if an accepting state is visited infinitely often (satisfies the Büchi condition) and satisfies the Parikh condition once.

Quite surprisingly, both models turn out to capture exactly the class $\mathcal{L}^{\omega}_{\mathsf{PA},\mathsf{Reg}}$, and hence are equivalent.

We then study strong reset Parikh automata (SPBA), which were introduced by Grobler et al. [6]. We consider the automata as directed graphs and provide two graph theoretic definitions of subclasses of SPBA that exactly capture the classes $\mathcal{L}_{\mathsf{Reg},\mathsf{PA}}^{\omega}$ and $\mathcal{L}_{\mathsf{PA},\mathsf{PA}}^{\omega}$. These definitions are based on an analysis of the strongly connected components of the underlying graph, where the accepting states can be found and how they are connected to the rest of the graph.

We believe that our results provide interesting insights into the theory of Parikhrecognizable ω -languages. It remains an interesting open question to characterize the new classes of ω -languages by logics.

2. Preliminaries

2.1. Finite and infinite words

We write \mathbb{N} for the set of non-negative integers including 0. 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 $L^{\omega} = \{w_1 w_2 \dots \mid w_i \in L \setminus \{\varepsilon\}\} \subseteq \Sigma^{\omega}$.

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 $L(\mathcal{A}) = \{w \in \Sigma^* \mid \text{there is an accepting run of } \mathcal{A} \text{ on } w\}$. If a language L is recognized by some NFA \mathcal{A} , we call L regular.

A Büchi Automaton (BA) 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 $\alpha_1\alpha_2\alpha_3\ldots$ is an infinite sequence of transitions $r = r_1r_2r_3\ldots$ 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 such that $p_i \in F$. We define the ω -language recognized by \mathcal{A} as $L_{\omega}(\mathcal{A}) = \{\alpha \in \Sigma^{\omega} \mid \text{there is an accepting run of } \mathcal{A} \text{ on } \alpha\}$. If an ω -language L is recognized by some BA \mathcal{A} , we call L ω -regular. Büchi's theorem establishes an important connection between regular and ω -regular languages:

Theorem 2.1 (Büchi). A language $L \subseteq \Sigma^{\omega}$ is ω -regular if and only if there are regular languages $U_1, V_1, \ldots, U_n, V_n \subseteq \Sigma^*$ for some $n \ge 1$ such that $L = U_1 V_1^{\omega} \cup \cdots \cup U_n V_n^{\omega}$.

2.3. Semi-linear sets

A linear set of dimension d for $d \ge 1$ is a set of the form $\{b_0 + b_1z_1 + \cdots + b_\ell z_\ell \mid z_1, \dots, z_\ell \in \mathbb{N}\} \subseteq \mathbb{N}^d$ for $b_0, \dots, b_\ell \in \mathbb{N}^d$. A semi-linear set is the finite union of linear sets. For vectors $\mathbf{u} = (u_1, \dots, u_c) \in \mathbb{N}^c$, $\mathbf{v} = (v_1, \dots, v_d) \in \mathbb{N}^d$, 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 the set of labeled transitions, and $C \subseteq \mathbb{N}^d$ is a semilinear 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 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 $1 \le i < k$. Similarly, a 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 $1 \le i < k$, 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 $U \in V(G)$ 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 relevant definitions of Parikh automata operating on infinite words introduced by Grobler et al. [6] and Guha et al. [7]. We then propose further definitions and compare the resulting automata.

A Parikh-Büchi automaton (PBA) is a PA $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$. A run of \mathcal{A} on an infinite word $\alpha = \alpha_1 \alpha_2 \alpha_3 \dots$ is an infinite sequence of labeled transitions $r = r_1 r_2 r_3 \dots$ 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 .

Let us first recall the definition of (strong) reset PBA (SPBA) introduced by Grobler et al. [6]: let $k_0 = 0$ and denote by k_1, k_2, \ldots the positions of all accepting states in r. Then r is accepting if k_1, k_2, \ldots is an infinite sequence and $\rho(r_{k_{i-1}+1} \ldots r_{k_i}) \in C$ for all $i \geq 1$. The ω -language recognized by an SPBA \mathcal{A} is $S_{\omega}(\mathcal{A}) = \{\alpha \mid \mathcal{A} \text{ accepts } \alpha\}$. Intuitively worded, whenever an SPBA enters an accepting state, the Parikh condition must be satisfied. Then the counters are reset.

Let us now recall the definition of (synchronous) reachability Parikh automata (RPA) introduced by Guhe et al. [7]. The run r is accepting 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 . The ω -language recognized by an RPA \mathcal{A} is $R_{\omega}(\mathcal{A}) = \{\alpha \mid \mathcal{A} \text{ accepts } \alpha\}$.

Let us finally recall the definition of prefix PBA (PPBA) introduced by Grobler et al. [6], which are obviously equivalent to (synchronous) Büchi Parikh automata introduced by Guhe et al. [7]. The run r is accepting if there are infinitely many $i \geq 1$ such that $p_i \in F$ and $\rho(r_1 \dots r_i) \in C$. The ω -language recognized by a PPBA \mathcal{A} is $P_{\omega}(\mathcal{A}) = \{\alpha \mid \mathcal{A} \text{ accepts } \alpha\}$. Hence, a PPBA can be seen as a stronger variant of RPA where we require infinitely many accepting hits instead of a single one.

We remark that we stick to the notation of the original papers, that is, we abbreviate (strong) reset Parikh-Büchi automata by SPBA and (synchronous) reachability Parikh automata by RPA. The attentive reader may have noticed that we do not use the term RPBA. The reason for this (in addition to sticking to the original notation) is that, unlike Büchi automata, RPA do *not* need to see an accepting state infinitely often. We show that this property implies that there are ω -regular languages that are not RPA-recogniazble. This motivates the study of RPA that additionally need to satisfy the Büchi-condition (which we hence will call RPBA).

We begin with a simple lemma. A similar result was proved in Theorem 3 of [7], however, RPA in [7] are assumed to be *complete*, i.e., for every state and every symbol there is at least one transition. The proof presented in [7] does not go through for the more general setting of non-complete RPA.

Lemma 3.1. There is an ω -regular language that is not recognized by any RPA.

Proof. We show that $L = \{\alpha \in \{a,b\}^{\omega} \mid |\alpha|_a = \infty\}$ is not RPA-recognizable. Assume that there is an RPA \mathcal{A} with $R_{\omega}(\mathcal{A}) = L$ and let n be the number of states of \mathcal{A} . As $\alpha = (a^nb^n)^{\omega} \in L$, there is an accepting run $r = r_1r_2r_3...$ of \mathcal{A} on α . Let i be the first position of an accepting hit in r. Now consider $\beta = (a^nb^n)^i \cdot b^{\omega} \notin L$. As α and β share the same prefix of length i, the run $r_1 ... r_i$ is also a partial run on $\beta[1,i]$, hence, generating an accepting hit. Now observe that in every infix of the form b^n a state is visited at least twice by the pigeonhole principle. Hence, we can "infinitely pump" a b-block after the accepting hit in r_i and obtain an accepting run on β , contradicting $R_{\omega}(\mathcal{A}) = L$.

Remark 3.1. In fact, complete RPA are strictly weaker than general RPA, as there is no complete RPA recognizing $\{a\}^{\omega}$ over $\Sigma = \{a,b\}$; an ω -language that is obviously RPA-recognizable.

As mentioned above, this weakness motivates the study of reachability Parikh-Büchi automata (RPBA). The run r is accepting if there is an $i \geq 1$ such that $\rho(r_1 \dots r_i) \in C$ and $p_i \in F$, and if there are infinitely many j such that $p_j \in F$. We define the ω -language recognized by an RPBA \mathcal{A} as $B_{\omega}(\mathcal{A}) = \{\alpha \mid \mathcal{A} \text{ accepts } \alpha\}$.

Every ω -regular language is RPBA-recognizable, as we can turn an arbitrary Büchi automaton into an equivalent RPBA by labeling every transition with 0 and setting $C = \{0\}$.

Finally, we define a variant of PBA motivated by a proposal of Klaedke and Ruess [8]. Here we consider semi-linear sets over $(\mathbb{N} \cup \{\infty\})^d$ and compute the extended Parikh image of an infinite run using transfinite induction. A limit Parikh-Büchi automaton (LPBA) is a PA $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ where C may use the symbol ∞ . The run r is accepting if there are infinitely many $i \geq 1$ such that $p_i \in F$, and if additionally $\rho(r) \in C$, where the j-th component of $\rho(r)$ is computed as follows. If there are infinitely many $i \geq 1$

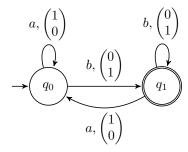


Figure 1: The automaton \mathcal{A} with $C = \{(z, z), (z, \infty) \mid z \in \mathbb{N}\}$ from Example 1.

such that the j-th component of \mathbf{v}_i has a non-zero value, then the j-th component of $\rho(C)$ 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 define the ω -language recognized by an LPBA \mathcal{A} as $L_{\omega}(\mathcal{A}) = \{\alpha \mid \mathcal{A} \text{ accepts } \alpha\}$.

Schnexample 1. Let \mathcal{A} be the automaton in Figure 1 with $C = \{(z, z), (z, \infty) \mid z \in \mathbb{N}\}.$

- If we interpret \mathcal{A} as a PA (over finite words), then we have $L(\mathcal{A}) = \{w \in \{a, b\}^* \cdot \{b\} \mid |w|_a = |w|_b\}$. The automaton is in the accepting state after reading a b. The first counter counts the number of read as, the second one reads the number of read bs. By definition of C the automaton only accepts when both counters are equal (note that vectors containing an ∞ -entry have no additional effect).
- If we interpret \mathcal{A} as an SPBA, then we have $S_{\omega}(\mathcal{A}) = \{ab\}^{\omega}$. Whenever the automaton reaches an accepting state also the Parikh condition must be satisfied. In the example this is only possible after reading exactly one a and one b. After that the counters are reset.
- If we interpret \mathcal{A} as a PPBA, then we have $P_{\omega}(\mathcal{A}) = L(\mathcal{A})^{\omega}$. The automaton accepts a word if infinitely often the Parikh condition is satisfied in the accepting state. Observe that C has no base vector and the initial state as well as the accepting state have the same outgoing edges.
- If we interpret \mathcal{A} as an LPBA, 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 word contains only some finite number z of as (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 an RPA, 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 $L(\mathcal{A})$. Since the automaton is complete it cannot get stuck and accepts any continuation after that.

• If we interpret \mathcal{A} as an RPBA, then we have $B_{\omega}(\mathcal{A}) = \{\alpha \in \{a,b\}^{\omega} \mid \alpha \text{ has a prefix in } L(\mathcal{A}) \text{ and } |\alpha|_b = \infty\}$. After having met the reachability condition the automaton still needs to satisfy the Büchi condition, which enforces infinitely many visits of the accepting state.

Remark 3.2. The automaton \mathcal{A} in the last example is deterministic. We note that $L_{\omega}(\mathcal{A})$ is not deterministic ω -regular but deterministic LPBA-recognizable.

We denote the class of ω -languages recognized by an SPBA (PPBA, RPA, RPBA, LPBA) by $\mathcal{L}_{\mathsf{SPBA}}$ ($\mathcal{L}_{\mathsf{PPBA}}$, $\mathcal{L}_{\mathsf{RPA}}$, $\mathcal{L}_{\mathsf{RPBA}}$, $\mathcal{L}_{\mathsf{LPBA}}$). Furthermore, denote by $\mathcal{L}_{\mathsf{PA},\mathsf{PA}}^{\omega}$ the class of ω -languages of the form $\bigcup_i U_i V_i^{\omega}$, where the U_i and V_i are Parikh-recognizable, by $\mathcal{L}_{\mathsf{PA},\mathsf{Reg}}^{\omega}$ such languages where the U_i are Parikh-recognizable and the V_i are regular, and by $\mathcal{L}_{\mathsf{Reg},\mathsf{PA}}^{\omega}$ such languages where the U_i are regular and the V_i are Parikh-recognizable. As shown by Guhe et al. we have $\mathcal{L}_{\mathsf{RPA}} \subsetneq \mathcal{L}_{\mathsf{PPBA}}$ [7]. Likewise, Grobler et al. [6] have shown $\mathcal{L}_{\mathsf{PPBA}} \subsetneq \mathcal{L}_{\mathsf{PA},\mathsf{PA}} \subsetneq \mathcal{L}_{\mathsf{SPBA}}$. We conclude this section by showing $\mathcal{L}_{\mathsf{RPA}} \subsetneq \mathcal{L}_{\mathsf{RPBA}} \subsetneq \mathcal{L}_{\mathsf{PPBA}}$. In the next section we show that $\mathcal{L}_{\mathsf{RPBA}} = \mathcal{L}_{\mathsf{LPBA}} = \mathcal{L}_{\mathsf{PA},\mathsf{Reg}}^{\omega}$.

Lemma 3.2. $\mathcal{L}_{RPA} \subsetneq \mathcal{L}_{RPBA}$.

Proof. We show $\mathcal{L}_{\mathsf{RPA}} \subseteq \mathcal{L}_{\mathsf{RPBA}}$. Strictness follows from Lemma 3.1. The proof is very similar to the proof that $\mathcal{L}_{\mathsf{RPA}} \subsetneq \mathcal{L}_{\mathsf{PPBA}}$ [7, Theorem 3]. Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ be an RPA. The idea is to create two copies of \mathcal{A} , where the second copy is modified such that all counters are zero and all states are accepting. Then we use non-determinism to guess the accepting hit and transition into the second copy where the counters are frozen and all states are accepting. Let $\mathcal{A}' = (Q', \Sigma, q_0, \Delta', F', C)$ where $Q' = \{q, q' \mid q \in Q\}$, $F' = \{q' \mid q \in Q\}$ and $\Delta' = \Delta \cup \{(p', a, \mathbf{0}, q') \mid (p, a, \mathbf{v}, q) \in \Delta\} \cup \{(p, a, \mathbf{v}, q') \mid (p, a, \mathbf{v}, q) \in \Delta\}$ be a RPBA. We claim that $R_{\omega}(\mathcal{A}) = B_{\omega}(\mathcal{A}')$.

 \Rightarrow To show $R_{\omega}(\mathcal{A}) \subseteq B_{\omega}(\mathcal{A}')$, let $\alpha \in R_{\omega}(\mathcal{A})$ 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 $i \geq 1$ be an arbitrary position such that there is an accepting hit in r_i (which exists by definition). Let $r'_i = (p_{i-1}, \alpha_i, \mathbf{0}, p'_i)$ and define $r'_j = (p'_{j-1}, \alpha_j, \mathbf{0}, p'_j)$ for all j > i. Then $r' = r_1 r_2 \dots r_{i-1} r'_i r'_{i+1} r'_{i+2} \dots$ is a run of \mathcal{A}' on α . Furthermore, r' is accepting: the accepting hit in r_i translates one-to-one to an accepting hit in r'_i . Furthermore, all p_j for $j \geq i$ are accepting by the definition of F'. Hence, r' is accepting, thus $\alpha \in B_{\omega}(\mathcal{A}')$.

 \Leftarrow To show $B_{\omega}(\mathcal{A}') \subseteq R_{\omega}(\mathcal{A})$, let $\alpha \in B_{\omega}(\mathcal{A}')$ with accepting run $r' = r'_1 r'_2 r'_3 \dots$ where $r'_i = (\hat{p}_{i-1}, \alpha_i, \mathbf{v}_i, \hat{p}_i)$ with $\hat{p}_i \in \{p_i, p'_i\}$. Again, let $i \geq 1$ be an arbitrary position such that there is an accepting hit in r'_i . As there are no accepting states in the first copy of \mathcal{A} in \mathcal{A}' , there is a point where r transitions from the first copy to the second copy, i. e., there is a $j \leq i$ such that $r'_j = (p_{j-1}, \alpha_j, \mathbf{v}_j, p'_j)$. Observe that the counters are frozen after transitioning to the second copy, hence, we have $\rho(r'_1 \dots r'_i) = \rho(r'_1 \dots r'_j) \in C$. Furthermore, we have $p_j \in F$ by the choice of Δ' . Finally, observe that $\hat{p}_\ell = p_\ell$ for all $\ell < j$, and $\hat{p}_\ell = p'_\ell$ for all $\ell \geq j$. Hence, we can replace r'_j by $r_j = (p_{j-1}, \alpha_j, \mathbf{v}_j, p_j)$, and for all $\ell > j$ we replace r'_ℓ by $r_\ell = (p_{\ell-1}, \alpha_\ell, \mathbf{v}_\ell, p_\ell)$, where \mathbf{v}_ℓ is arbitrary such that $r_\ell \in \Delta$ (observe that at least one such \mathbf{v} exists by definition of Δ'). Then $r = r'_1 r'_2 \dots r'_{j-1} r_j r_{j+1} r_{j+2} \dots$ is a run of \mathcal{A} on α that is furthermore accepting as witnessed by the accepting hit in r_j . Hence $\alpha \in R_\omega(\mathcal{A})$.

Observe that a very similar construction can be used to turn an arbitrary RPBA into an equivalent PPBA. The only difference is that we choose $F' = \{q' \mid q \in F\}$. Hence we obtain the following corollary.

Corollary 3.1. $\mathcal{L}_{\mathsf{RPBA}} \subseteq \mathcal{L}_{\mathsf{PPBA}}$.

Finally, we show that this inclusion is also strict.

Lemma 3.3. There is an ω-language that is PPBA-recognizable but not RPBA-recognizable.

Proof. Consider $L = \{\alpha \in \{a,b\}^{\omega} \mid |\alpha[1,i]|_a = |\alpha[1,i]|_b$ for infinitely many $i\}$, which is obviously PPBA-recognazable.

Assume that L is recognized by an RPBA \mathcal{A} and let n be the number of states of \mathcal{A} . Consider an accepting run $r = r_1 r_2 r_3 \dots$ of \mathcal{A} on $\alpha = (a^n b^n)^\omega$ where $r_i = (p_{i-1}, \alpha_i, \mathbf{v}_i, p_i)$, and let k be the position of the accepting hit, i. e. $p_k \in F$ and $\rho(r_1 \dots r_k) \in C$. By definition there are infinitely many $j \geq 1$ (and hence infinitely many $j \geq k$) such that $p_j \in F$. By the pigeonhole principle, there is a state q that is visited twice while reading an arbitrary a^n -infix, say at positions $k \leq c < d$, i. e., $p_c = p_d = q$ and d - c < n. Hence, we can pump an infix of the form a^{d-c} and obtain an accepting run on an infinite word of the form $(a^n b^n)^* (a^{n+d-c} b^n)(a^n b^n)^\omega$, which is not in L, a contradiction.

4. Characterization of $\mathcal{L}^{\omega}_{\mathsf{PA},\mathsf{Reg}}$ by limit PBA and reachability PBA

The main goal of this section is to prove the following theorem.

Theorem 4.1. 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 $U_i \in \Sigma^*$ is Parikh-recognizable, and $V_i \subseteq \Sigma^*$ is regular.
- 2. L is LPBA-recognizable.
- 3. L is RPBA-recognizable.

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 $\bigcup_i U_i V_i = \bigcup_i U_i (\bigcup_{j_i} X_{j_i} Y_{j_i}^\omega) = \bigcup_{i,j_i} U_i (X_{j_i} Y_{j_i}^\omega) = \bigcup_{i,j_i} U_i (X_{j_i} Y_{j_i}^\omega) = \bigcup_{i,j_i} U_i (X_{j_i} Y_{j_i}^\omega)$, for regular languages X_{j_i}, Y_{j_i} , where $U_i X_{j_i}$ is regular, as regular languages are closed under concatenation.

To simplify the proof, it is convenient to consider the following generalizations of Büchi automata. A generalized Büchi automaton (GBA) is a tuple $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{F})$ where Q, Σ, q_0 and Δ are defined as for BA, and $\mathcal{F} \subseteq 2^Q$ is a collection of sets of accepting states. Then a run $r_1r_2r_3...$ with $r_i = (p_{i-1}, \alpha_i, p_i)$ is accepting if for all $F \in \mathcal{F}$ there are infinitely many i such that $p_i \in \mathcal{F}$. It is well-known that GBA are not more expressive than BA, see e.g. [2, Theorem 4.56].

Furthermore, we consider a variant of GBA where acceptance is not defined via states that are seen infinitely often, but rather via transitions that are used infinitely often. A Generalized Transition Büchi Automaton (GTBA) 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 $r_1r_2r_3...$ is accepting if for all $T \in \mathcal{T}$ there are infinitely many i such that $r_i \in T$.

Lemma 4.1. GTBA and BA have the same expressiveness.

Proof. Obviously, every BA can be turned into an equivalent GTBA by choosing $\mathcal{T} = \{\{(p, a, q) \mid q \in F\}\}\$, hence we focus on the other direction.

Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{T})$ be a GTBA. As GBA are as expressive as BA, it is sufficient to convert \mathcal{A} into an equivalent GBA. The idea is basically to consider the line graph of \mathcal{A} , that is, to use Δ as the new state set, keeping the initial state. Then there is a b-transition from the state (p, a, q) to every state of the form (q, b, t). Hence, the acceptance component translates directly. To be precise, we construct the GBA $\mathcal{A}' = (\Delta \cup \{q_0\}, \Sigma, q_0, \Delta', \mathcal{T})$ where $\Delta' = \{((p, a, q), b, (q, b, t) \mid (p, a, q), (q, b, t) \in \Delta\} \cup \{(q_0, a, (q_0, a, q) \mid (q_0, a, q) \in \Delta\}$. It is now easily verified that $L_{\omega}(\mathcal{A}) = L_{\omega}(\mathcal{A}')$.

Theorem 4.1 will be a direct consequence from the following lemmas. The first lemma shows the implication $(1) \Rightarrow (2)$.

Lemma 4.2. If $L \in \mathcal{L}^{\omega}_{\mathsf{PA},\mathsf{Reg}}$, then L is LPBA-recognizable.

Proof. First observe that $\mathcal{L}_{\mathsf{LPBA}}$ is closed under union (this can be shown using a standard construction). Hence, it is sufficient to show how to construct an LPBA 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 an LPBA 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 $r_1r_2r_3\ldots$ 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 $\rho(r) = \rho(r_1 \ldots r_{j-1}) \in C$ (in particular, there is no ∞ value in $\rho(r)$). We observe that $r_1 \ldots r_{j-1}$ is an accepting run of \mathcal{A}_1 on $\alpha[1, j-1]$, as $p_{j-1} \in F_1$ and $\rho(r_1 \ldots r_{j-1}) \in C$. For all $i \geq j$ let $r'_i = (p_{i-1}, \alpha_i, p_i)$. Now observe that $(q_2, \alpha_j, p_j)r'_{j+1}r'_{j+2}\ldots$ is an accepting run of \mathcal{A}_2 on $\alpha_j\alpha_{j+1}\alpha_{j+2}\ldots$, 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 $sr'_1 r'_2 r'_3 \dots$ 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 an RPBA (every visit of an accepting state has the same good counter value; this argument is even true if we interpret \mathcal{A} as a PPBA), showing the implication (1) \Rightarrow (3).

Corollary 4.1. If $L \in \mathcal{L}^{\omega}_{\mathsf{PA},\mathsf{Reg}}$, then L is RPBA-recognizable.

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 4.3. 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 by the following procedure.

- 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 $\mathbf{v} = (v_1, \dots, v_n)$ 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 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 $C_D = \varnothing$. 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 $(2) \Rightarrow (1)$.

Lemma 4.4. If L is LPBA-recognizable, then $L \in \mathcal{L}_{\mathsf{PA},\mathsf{Reg}}^{\omega}$.

Proof. Let $\mathcal{A}=(Q,\Sigma,q_0,\Delta,F,C)$ be an LPBA of dimension d. The idea is as follows. We guess a subset $D\subseteq\{1,\ldots,d\}$ 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 $D\subseteq\{1,\ldots,d\}$ we decompose $\mathcal A$ into a PA and a GTBA. 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 GTBA 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 GTBA. Furthermore we delete all transitions that modify counters not in D.

Fix $D \subseteq \{1,\ldots,d\}$ and $f \in F$, and define the PA $\mathcal{A}_f^D = (Q,\Sigma,q_0,\Delta,\{f\},C_D)$ where C_D is defined as in Lemma 4.3. Furthermore, we define the GTBA $\mathcal{B}_f^D = (Q,\Sigma,f,\Delta^D,\mathcal{T}^D)$ 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 GTBA). 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 i-th counter has a non-zero value. Finally, we encode the condition that at least one accepting state in F needs to by seen in \mathcal{T}^D by further adding the component $\{(p,a,q)\in\Delta\mid q\in F\}$.

We claim that $L_{\omega}(\mathcal{A}) = \bigcup_{D \subseteq \{1,\dots,d\}, f \in F} L(\mathcal{A}_f^D) \cdot L_{\omega}(\mathcal{B}_f^D)$, which by the comment below Theorem 4.1 and Lemma 4.1 implies the statement of the lemma.

 \Rightarrow To show $L_{\omega}(\mathcal{A}) \subseteq \bigcup_{D \subseteq \{1,\ldots,d\},f \in F} L(\mathcal{A}_f^D) \cdot L_{\omega}(\mathcal{B}_f^D)$, let $\alpha \in L_{\omega}(\mathcal{A})$ with accepting run $r_1r_2r_3\ldots$ where $r_i = (p_{i-1},\alpha_i,\mathbf{v}_i,p_i)$. Let D be the positions of ∞ -entries in $\rho(r) = (v_1,\ldots,v_d)$. As the v_i with $i \notin D$ have integer values, there is a position j such that in all \mathbf{v}_k for $k \geq j$ the i-th entry of \mathbf{v}_k is 0. Let $\ell \geq j$ be minimal such that p_ℓ in F. We split $\alpha = w\beta$, where $w = \alpha[1,\ell]$, and $\beta = \alpha_{\ell+1}\alpha_{\ell+2}\ldots$

We split $\alpha = w\beta$, where $w = \alpha[1,\ell]$, and $\beta = \alpha_{\ell+1}\alpha_{\ell+2}\dots$ First we argue that $w \in L_{\omega}(\mathcal{A}_{p_{\ell}}^{D})$. Observe that $\mathcal{A}_{p_{\ell}}^{D}$ inherits all transitions from \mathcal{A} , hence $r_1 \dots r_{\ell}$ is a run of $\mathcal{A}_{p_{\ell}}^{D}$ on w. As p_{ℓ} is accepting by definition, it remains to show that $\rho(r_1, \dots r_{\ell}) \in C_D$. By the choice of ℓ , all counters not in D have reached their final values. As C_D contains all vectors of C where all ∞ -entries are replaced by arbitrary values, the claim follows, hence $w \in L(\mathcal{A}_{p_{\ell}}^{D})$.

Now we argue that $\beta \in L_{\omega}(\mathcal{B}_{p_{\ell}}^{D})$. For every $k > \ell$ define $r'_{k} = (p_{k-1}, \alpha_{k}, p_{k})$. Observe that $r' = r'_{k+1}r'_{k+2}\dots$ is a run of $\mathcal{B}_{p_{\ell}}^{D}$ on β (all r'_{k+1} exist in $\mathcal{B}_{p_{\ell}}^{D}$, as the counters not on D of all transitions r_{k} have zero-values by the definition of ℓ). It remains to show that r' is accepting, i. e., that for every counter in D at least one transition with a non-zero values is used infinitely often, and an accepting state is visited infinitely often. This is the case, as these counter values are ∞ in $\rho(r)$ and by the acceptance condition of LPBA, hence $\beta \in L_{\omega}(\mathcal{B}_{p_{\ell}}^{D})$.

We conclude $\alpha \in \bigcup_{D \subseteq \{1,\dots,d\}, f \in F} L(\mathcal{A}_f^D) \cdot L_{\omega}(\mathcal{B}_f^D)$.

 $\Leftarrow \text{ To show } \bigcup_{D\subseteq\{1,\ldots,d\},f\in F} L(\mathcal{A}_f^D) \cdot L_{\omega}(\mathcal{B}_f^D) \subseteq L_{\omega}(\mathcal{A}), \text{ let } w\in L(\mathcal{A}_f^D) \text{ and } \beta\in L_{\omega}(\mathcal{B}_f^D)$ for some $D\subseteq\{1,\ldots,d\}$ and $f\in F$. We show that $w\beta\in L_{\omega}(\mathcal{A})$.

Let s be an accepting run of \mathcal{A}_f^D on w, which ends in the accepting state f by definition. Let $\rho(s)=(v_1,\ldots,v_d)$. By definition of C_D , there is a vector $\mathbf{u}=(u_1,\ldots,u_d)$ in C where $u_i=\infty$ if $i\in D$, and $u_i=v_i$ if $i\notin D$. Furthermore, let $r=r_1r_2r_3\ldots$, where $r_i=(p_{i-1},\alpha_i,p_i)$, be an accepting run of \mathcal{B}_f^D on β , which starts in the accepting state f by definition. By definition of \mathcal{T}^d , for every counter $i\in D$ at least one transition where the i-th counter of the corresponding transition in Δ is non-zero is used infinitely often. Hence, let $r'=r'_1r'_2r'_3\ldots$ where $r'_i=(p_{i-1},\alpha_i,\mathbf{v}_i,p_i)$ for a suitable vector \mathbf{v}_i . Furthermore, the labels of transitions of counters not in D have a value of zero, hence $\rho(r')=(x_1,\ldots,x_d)$, where $x_i=\infty$ if $i\in D$, and $x_i=0$ if $i\notin D$. A technical remark: it might be the case that there are more than one transitions in Δ that collapse to the same transition in Δ^D , say $\delta_1=(p,a,\mathbf{u},q)$ and $\delta_2=(p,a,\mathbf{v},q)$ appear in Δ and collapse to (p,a,q) in Δ^D . If both transitions, δ_1 and δ_2 , are seen infinitely often, we need to take care that we also see both infinitely often when translating the run r back. This is possible using a round-robin procedure.

Now observe that sr' is a run of \mathcal{A} on $w\beta$ (recall that s ends in f, and r' starts in f). Furthermore, we have $\rho(sr') = \rho(s) + \rho(r') = (v_1 + x_1, \dots, v_d + x_d)$, where $v_i + x_i = \infty$ if $i \in D$, and $v_i + x_i = v_i$ if $i \notin D$ by the observations above. Hence $\rho(sr') \in C$. Finally, \mathcal{T}^D enforces that at least one accepting state in \mathcal{B}_f^D is seen infinitely often, hence $w\beta \in L_{\omega}(\mathcal{A})$.

Finally we show the implication $(3) \Rightarrow (1)$.

Lemma 4.5. If L is RPBA-recognizable, then $L \in \mathcal{L}_{\mathsf{PA},\mathsf{Reg}}^{\omega}$.

Proof. Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ be an RPBA. The intuition is as follows. An RPBA 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 $B_{\omega}(\mathcal{A}) = \bigcup_{f \in F} L(\mathcal{A}_f) \cdot L_{\omega}(\mathcal{B}_f)$.

 \Rightarrow To show $B_{\omega}(\mathcal{A}) \subseteq \bigcup_{f \in F} L(\mathcal{A}_f) \cdot L_{\omega}(\mathcal{B}_f)$, 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 $\bigcup_{f \in F} L(\mathcal{A}_f) \cdot L_{\omega}(\mathcal{B}_f) \subseteq B_{\omega}(\mathcal{A})$, 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 $w\beta \in B_{\omega}(\mathcal{A})$.

5. Characterization of $\mathcal{L}_{PA,PA}^{\omega}$ and $\mathcal{L}_{Reg,PA}^{\omega}$

In this section we give a characterization of $\mathcal{L}^{\omega}_{\mathsf{PA},\mathsf{PA}}$ and a characterization of $\mathcal{L}^{\omega}_{\mathsf{Reg},\mathsf{PA}}$. Grobler et al. [6] have shown that $\mathcal{L}^{\omega}_{\mathsf{PA},\mathsf{PA}} \subsetneq \mathcal{L}_{\mathsf{SPBA}}$, i. e., SPBA are too strong to capture this class. However, restrictions of SPBA are a good candidate to capture $\mathcal{L}^{\omega}_{\mathsf{PA},\mathsf{PA}}$ as well as $\mathcal{L}^{\omega}_{\mathsf{Reg},\mathsf{PA}}$. In fact we show that it is sufficient to restrict the appearances of accepting states to capture $\mathcal{L}^{\omega}_{\mathsf{PA},\mathsf{PA}}$, as specified by the first theorem of this section. Further restricting the vectors yields a model capturing $\mathcal{L}^{\omega}_{\mathsf{Reg},\mathsf{PA}}$, as specified in the second theorem of this section. Recall that the condensation of \mathcal{A} is the DAG of strong components of the underlying graph of \mathcal{A} .

Theorem 5.1. 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 an SPBA \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.

In fact, the proof of Grobler et al. [6] showing $\mathcal{L}_{PA,PA}^{\omega} \subseteq \mathcal{L}_{SPBA}$ is constructive and almost yields an SPBA with the desired property. A key notion are normalized PA (on finite words), where a PA $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ is normalized if $F = \{f\}$ and f has no outgoing transitions. It was shown that one can, given a PA \mathcal{A} , construct a normalized PA A_N with $L(A_N) = L(A) \setminus \{\varepsilon\}$. For our proofs it is convenient to introduce a similar, yet stronger notion.

Lemma 5.1. Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ be a PA of dimension d. Then there exists a PA \mathcal{A}^{IO} of dimension d+1 with the following properties.

- \bullet The initial state of \mathcal{A}^{IO} is the only accepting state.
- $L(\mathcal{A}) \setminus \{\varepsilon\} = L(\mathcal{A}^{IO}) \setminus \{\varepsilon\}.$ $SCC(\mathcal{A}) = \{Q\}.$

Proof. Define
$$\mathcal{A}^{IO} = \{Q \cup \{q'_0\}, \Sigma, q'_0, \Delta^{IO}, \{q'_0\}, C \cdot \{1\}), \text{ where}$$

$$\Delta^{IO} = \{(p, a, \mathbf{v} \cdot 0, q) \mid (p, a, \mathbf{v}, q) \in \Delta\}$$

$$\cup \{(q'_0, a, \mathbf{v} \cdot 0, q) \mid (q_0, a, \mathbf{v}, q) \in \Delta\}$$

$$\cup \{(p, a, \mathbf{v} \cdot 1, q'_0) \mid (p, a, \mathbf{v}, f) \in \Delta, f \in F\}$$

$$\cup \{(q'_0, a, \mathbf{v} \cdot 1, q'_0) \mid (q_0, a, \mathbf{v}, f) \in \Delta, f \in F\}.$$

That is, \mathcal{A}^{IO} is obtained from \mathcal{A} by adding a fresh state q'_0 , which is the initial state and only accepting state, inherits all outgoing transitions from q_0 and all in-going transitions from the accepting states. Furthermore, all transitions get a new counter, which is set to 0 except for the new ingoing transitions of q'_0 where the counter is set to 1, and all vectors in C are concatenated with 1. Finally, we remove all states that cannot reach q'_0 (such states can appear when shortcutting the ingoing transitions of F, and are useless in the sense that their removal does not change the accepted language; however, this removal is necessary for the third property). We claim that $L(\mathcal{A})\setminus\{\varepsilon\}=L(\mathcal{A}^{IO})\setminus\{\varepsilon\}$.

 $\Rightarrow \text{To show } L(\mathcal{A}) \setminus \{\varepsilon\} \subseteq L(\mathcal{A}^{IO}) \setminus \{\varepsilon\}, \text{ let } w_1 \dots w_n \in L(\mathcal{A}) \text{ for } n \geq 1 \text{ with accepting } \text{run } r = r_1 \dots r_n \text{ where } r_i = (p_{i-1}, w_i, \mathbf{v}_i, p_i). \text{ By definition of } \Delta^{IO}, \text{ there is a transition } r'_1 = (q'_0, w_1, \mathbf{v}_1 \cdot 0, p_1) \text{ as well as a transition } r'_n = (p_{n-1}, w_n, \mathbf{v}_n \cdot 1, q'_0) \text{ (or in case } n = 1 \text{ the loop } (q'_0, w_1, \mathbf{v}_1 \cdot 1, q'_0)). \text{ For all } 1 < i < n \text{ define } r'_i = (p_{i-1}, w_i, \mathbf{v}_i \cdot 0, p_i). \text{ It is now easily verified that } r'_1 \dots r'_n \text{ (or simply } (q'_0, w_1, \mathbf{v}_1 \cdot 1, q'_0)) \text{ is an accepting run of } \mathcal{A}^{IO} \text{ on } \mathbf{v}^{IO} = \mathbf{v}^{IO} \mathbf{v}^{IO}$

 \Leftarrow To show $L(\mathcal{A}^{IO}) \setminus \{\varepsilon\} \subseteq L(\mathcal{A}) \setminus \{\varepsilon\}$, let $w_1 \dots w_n \in L(\mathcal{A}^{IO})$ for $n \geq 1$ with accepting run $r' = r'_1 \dots r'_n$ where $r'_i = (p_{i-1}, w_i, \mathbf{v}_i \cdot c_i, p_i)$ with $c_i \in \{0, 1\}$. Observe that $p_0 = p_n = q'_0$, and for all 0 < i < n we have $p_i \neq q'_0$ as enforced by the additional counter (that is, $c_n = 1$ and $c_i = 0$ for all i < n, as $C \cdot \{1\}$ is the semi-linear set of \mathcal{A}^{IO}). By definition of Δ^{IO} there is a transition $r_1 = (q_0, w_1, \mathbf{v}_1, p_1)$, and a transition $r_n = (p_{n-1}, w_n, \mathbf{v}_n, f)$ for some $f \in F$ in Δ (or in case n = 1 the transition $(q_0, w_1, \mathbf{v}_1, f)$). For all 1 < i < n define $r_i = (p_{i-1}, w_i, \mathbf{v}_i, p_i)$. It is now easily verified that $r_1 \dots r_n$ (or simply $(q_0, w_1, \mathbf{v}_1, f)$ is an accepting run of \mathcal{A} on $w_1 \dots w_n$.

Observe that $L(\mathcal{A})^{\omega} = L(\mathcal{A}^{IO})^{\omega}$ for every PA \mathcal{A} , as $L^{\omega} = (L \setminus \{\varepsilon\})^{\omega}$ for every language L by definition. In fact, it is easily observed that we even have $S_{\omega}(\mathcal{A}^{IO}) = L(\mathcal{A})^{\omega}$. We are now ready to proof the main theorem.

Proof of Theorem 5.1. (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 5.1 and the observation above we can equivalently write $L = L(\mathcal{A}_1) \cdot S_{\omega}(\mathcal{A}_2^{IO})$. As \mathcal{A}_2^{IO} is IO-normalized it satisfies the property of the theorem.

We can now easily adapt the construction in [6] showing that the concatenation of a Parikh-recognizable language and an SPBA-recognizable ω -language is SPBA-recognizable to obtain an SPBA for $L(A_1) \cdot S_{\omega}(A_2^{IO})$ that only keeps the accepting state of A_2^{IO} , maintaining the property of the theorem. Finally, the closure under union is shown using a standard construction, hence combining SPBA with the desired property still yields an SPBA with the property. Overall, we obtain an SPBA A recognizing L where the only accepting states appear in the leaves of C(A).

 $(2)\Rightarrow (1)$. Let $\mathcal{A}=(Q,\Sigma,q_0,\Delta,F,C)$ be an SPBA 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 of the ingoing 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} by setting f as the initial state and only accepting state, where Δ_f is defined as for \mathcal{A}_f . We claim $S_{\omega}(\mathcal{A})=\bigcup_{f\in F}L(\mathcal{A}_f)\cdot L(\mathcal{A}_{f,f})^{\omega}$.

 \Rightarrow To show $S_{\omega}(\mathcal{A}) \subseteq \bigcup_{f \in F} L(\mathcal{A}_f) \cdot L(\mathcal{A}_{f,f})^{\omega}$, 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 all $j \geq 1$ 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 $r_1
ldots r_k$ and observe that $p_i \neq p_{k_1}$ for all $i < k_1$, and $\rho(r_1
ldots r_{k_1}) \in C$ by the definition of SPBA. Hence $r' = r'_1
ldots r'_{k_1}$ 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 $\alpha[k_1+1,k_{i+1}]$: 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 $\alpha[k_1+1,k_{i+1}]\in L(\mathcal{A}_{p_{k_1},p_{k_1}})$. We conclude $\alpha\in L(\mathcal{A}_{p_{k_1}})\cdot L(\mathcal{A}_{p_{k_1},p_{k_1}})^{\omega}$.

 \Leftarrow To show $\bigcup_{f \in F} L(\mathcal{A}_f) \cdot L(\mathcal{A}_{f,f})^{\omega} \subseteq S_{\omega}(\mathcal{A})$, 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 $uv_1v_2 \dots \in S_{\omega}(\mathcal{A})$.

First let $u=u_1\ldots u_n$ and $r'=r'_1\ldots r'_n$ 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 $r_1\ldots r_n$ is a partial run of $\mathcal A$ on w with $\rho(r_1\ldots r_n)\in C$ 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, \ldots using the same argument. Hence $uv_1v_2\cdots \in S_{\omega}(\mathcal{A})$, concluding the proof.

As a side product of the proof of Theorem 5.1 we get the following corollary, which is in general not true for SPBA.

Corollary 5.1. Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ be an SPBA 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 $S_{\omega}(\mathcal{A}) = \bigcup_{f \in F} S_{\omega}(Q, \Sigma, q_0, \Delta, \{f\}, C)$.

By even further restricting the power of SPBA, we get the following characterization of $\mathcal{L}^{\omega}_{\mathsf{Re}\sigma,\mathsf{PA}}$.

Theorem 5.2. 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 Parikhrecognizable.
- 2. L is recognized by an SPBA \mathcal{A} with the following properties.
 - (a) At most one state q per leaf of the condensation of \mathcal{A} may have ingoing 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 5.1, hence, SPBA with this restriction are at most as powerful as those that characterize $\mathcal{L}_{\mathsf{PA},\mathsf{PA}}^{\omega}$. However, as a side product of the proof we get that property (a) is equivalent to the property of Theorem 5.1. Hence, property (b) is mandatory to sufficiently weaken SPBA such that they capture $\mathcal{L}_{\mathsf{Reg},\mathsf{PA}}^{\omega}$. In fact, using the notion of IO-normalization, we can re-use most of the ideas in the proof of Theorem 5.1.

Proof of Theorem 5.2. (1) \Rightarrow (2). We can trivially convert an NFA into an equivalent PA by labeling every transition with 0 and choosing $C = \{0\}$. Let \mathcal{A} be an arbitrary PA and observe that \mathcal{A}^{IO} has only a single SCC by definition. Again, we have $L(\mathcal{A})^{\omega} = S_{\omega}(\mathcal{A}^{IO})$ and the constructions for concatenation and union do not destroy the properties, hence we obtain an SPBA of the desired form.

- $(2) \Rightarrow (1) \text{ Let } \mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C) \text{ be an SPBA of dimension } d \text{ with properties (a)}$ and (b). Fix $f \in F$ and let $\mathcal{B}_f = (Q_f, \Sigma, q_0, \{(p, a, q) \mid (p, a, \mathbf{v}, q) \in \Delta, p, q \in Q_f\}, \{f\})$ with $Q_f = \{q \in Q \mid q \text{ appears in a non-leaf SCC of } C(\mathcal{A})\} \cup \{f\}$ be the NFA obtained from \mathcal{A} by removing all leaf states except f, and removing all labels from the transitions. Recycling the automaton from Theorem 5.1, let $\mathcal{A}_{f,f} = (Q, \Sigma, f, \Delta_f, \{f\}, C \cdot \{1\})$ with $\Delta_f = \{p, a, \mathbf{v} \cdot 0, q) \mid (p, a, \mathbf{v}, q) \in \Delta, q \neq f\} \cup \{(p, a, \mathbf{v} \cdot 1, f) \mid (p, a, \mathbf{v}, f) \in \Delta\}$. We claim $S_{\omega}(\mathcal{A}) = \bigcup_{f \in F} L(\mathcal{B}_f) \cdot L(\mathcal{A}_f)^{\omega}$.
- \Rightarrow To show $S_{\omega}(\mathcal{A}) = \bigcup_{f \in F} L(\mathcal{B}_f) \cdot L(\mathcal{A}_{f,f})^{\omega}$, let $\alpha \in S_{\omega}(\mathcal{A})$ with accepting run $r = r_1 r_2 r_3 \dots$ where $r_i = (p_{i-1}, \alpha_i, \mathbf{v}_i, p_i)$, and let $k_1 < k_2 < \dots$ be the positions of the accepting states in r, and consider the partial run $r_1 \dots r_{k_1}$ (if $k_1 = 0$, i.e., the initial state is already accepting, then $r_1 \dots r_{k_1}$ is empty).

By property (a) we have that p_{k_1} is the first state visited in r that is located in a leaf of C(A). Hence $r'_1 \ldots r'_{k_1}$, where $r'_i = (p_{i-1}, \alpha_i, p_i)$, is an accepting run of $\mathcal{B}_{p_{k_1}}$ on $\alpha[1, k_1]$ (in the case $k_1 = 0$ we define $\alpha[1, k_1] = \varepsilon$).

By the same argument as in the proof of Theorem 5.1 we have $p_{k_i} = p_{k_j}$ for all $i, j \ge 1$, hence $\alpha[k_i+1, k_{i+1}] \in L(\mathcal{A}_{p_{k_1}, p_{k_1}})$, and hence $\alpha \in L(\mathcal{B}_{p_k}) \cdot L(\mathcal{A}_{p_{k_1}, p_{k_1}})^{\omega}$.

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

First observe that properties (a) and (b) enforce that $\mathbf{0} \in C$, as the accepting state of a leaf of $C(\mathcal{A})$ is visited before a transition labeled with a non-zero can be used. Let $u=u_1\ldots u_n$ and $s_1\ldots s_n$ with $s_i=(p_{i_1},u_i,p_i)$ be an accepting run of \mathcal{B}_f on u. Define $s_i'=(p_{i_1},u_i,\mathbf{0},p_i)$ and observe that $s_1'\ldots s_n'$ is a partial run of \mathcal{A} with $\rho(s_1'\ldots s_n')\in C$ and $p_n=f$ by the observation above.

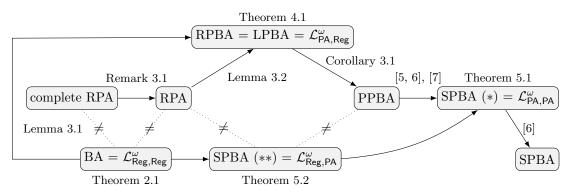
Again we can very similarly continue the run on v_1, v_2, \ldots using the same argument. Hence $uv_1v_2\cdots \in S_{\omega}(\mathcal{A})$, concluding the proof.

6. Conclusion

We conclude with an overview of our results shown in Figure 2. To finalize the picture, we observe the following.

 $Observation\ 6.1.$

- 1. There are RPA-recognizable ω -languages not contained in $\mathcal{L}^{\omega}_{\mathsf{Reg},\mathsf{PA}}$, for example $\{a^nb^n\mid n\geq 0\}\{a\}^{\omega}$.
- 2. We have $\mathcal{L}^{\omega}_{\mathsf{Reg},\mathsf{PA}} \subseteq \mathcal{L}^{\omega}_{\mathsf{PA},\mathsf{PA}}$ by definition, and there are ω -languages contained in $\mathcal{L}^{\omega}_{\mathsf{PA},\mathsf{PA}}$ that are not contained in $\mathcal{L}^{\omega}_{\mathsf{Reg},\mathsf{PA}}$, for example $\{a^nb^n \mid n \geq 1\}\{c^nd^n \mid n \geq 1\}^{\omega}$.
- 3. We have $\mathcal{L}^{\omega}_{\mathsf{Reg},\mathsf{Reg}} \subseteq \mathcal{L}^{\omega}_{\mathsf{Reg},\mathsf{PA}}$ by definition, and there are ω -languages contained in $\mathcal{L}^{\omega}_{\mathsf{Reg},\mathsf{PA}}$ that are neither PPBA-recognizable nor ω -regular, as witnessed by many ω -closures of Parikh-recognizable languages (which are trivially contained in $\mathcal{L}^{\omega}_{\mathsf{Reg},\mathsf{PA}}$), for example $\{a^nb^n\mid n\geq 1\}^{\omega}$ (a formal proof can be found for blind counter machines in [5], which are known to be equivalent to PPBA [6]).



- (*) At most one state q per leaf of C(A) may have ingoing 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 2: Overview of our results. Arrows mean strict inclusions and \neq means orthogonal.

Finally, we recall that deterministic ω -regular languages are characterized as regular arrow-languages \vec{L} , where $\vec{L} = \{\alpha \mid \alpha[1,i] \in L \text{ for infinitely many } i\}$ [10]. This characterization can easily be adapted to show that deterministic PPBA-recognizable ω -languages are captured by arrows of deterministic Parikh-recognizable languages. We conjecture that PPBA-recognizable ω -languages are captured by arrows of Parikh-recognizable languages yielding a similar characterization of $\mathcal{L}_{\text{PPBA}}$.

References

- [1] Joël Allred and Ulrich Ultes-Nitsche. k-counting automata. RAIRO Theoretical Informatics and Applications Informatique Théorique et Applications, 46(4):461–478, 2012.
- [2] Christel Baier and Joost-Pieter Katoen. Principles of Model Checking. The MIT Press, 2008.
- [3] Mikolaj Bojanczyk. Beyond omega-Regular Languages. In Jean-Yves Marion and Thomas Schwentick, editors, 27th International Symposium on Theoretical Aspects of Computer Science, volume 5 of Leibniz International Proceedings in Informatics (LIPIcs), pages 11–16, Dagstuhl, Germany, 2010. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- [4] J. Richard Büchi. Weak second-order arithmetic and finite automata. Mathematical Logic Quarterly, 6(1-6):66-92, 1960.
- [5] Henning Fernau and Ralf Stiebe. Blind counter automata on omega-words. Fundam. Inform., 83:51-64, 2008.
- [6] Mario Grobler, Leif Sabellek, and Sebastian Siebertz. Parikh automata on infinite words, 2023.
- [7] Shibashis Guha, Ismaël Jecker, Karoliina Lehtinen, and Martin Zimmermann. Parikh automata over infinite words, 2022.
- [8] Felix Klaedtke and Harald Rueß. Monadic second-order logics with cardinalities. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, Automata, Languages and Programming, pages 681–696, Berlin, Heidelberg, 2003. Springer.
- [9] Dario Della Monica, Angelo Montanari, and Pietro Sala. Beyond ωBS-regular languages: ωt-regular expressions and counter-check automata. Electronic Proceedings in Theoretical Computer Science, 256:223–237, 2017.
- [10] Wolfgang Thomas. Automata on Infinite Objects, page 133–191. MIT Press, Cambridge, MA, USA, 1991.