• RESEARCH PAPER •

On the complexity of ω-pushdown automata

Yusi LEI[1], Fu SONG[1,2*], Wanwei LIU[3] & Min ZHANG[1*]

[1]*Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China;*
[2]*School of Information Science and Technology, ShanghaiTech University, Shanghai 200031, China;*
[3]*School of Computer Science, National University of Defense Technology, Changsha 410073, China*

Abstract   Finite automata over infinite words (called ω-automata) play an important role in the automata-theoretic approach to system verification. Different types of ω-automata differ in their succinctness and complexity of their emptiness problems, as a result, theory of ω-automata has received considerable research attention. Pushdown automata over infinite words (called ω-PDAs), a generalization of ω-automata, are a natural model of recursive programs. Our goal in this paper is to conduct a relatively complete investigation on the complexity of the emptiness problems for variants of ω-PDAs. For this purpose, we consider ω-PDAs of five standard acceptance types: Büchi, Parity, Rabin, Streett and Muller acceptances. Based on the transformation for ω-automata and the efficient algorithm proposed by Esparza et al. in CAV'00 for verifying the emptiness problem of ω-PDAs with Büchi acceptance, it is trivial to check the emptiness problem of other ω-PDAs. However, this naive approach is not optimal. In this paper, we propose novel algorithms for the emptiness problem of ω-PDAs based on the observations of the structure of accepting runs. Our algorithms outperform algorithms that go through Büchi PDAs. In particular, the space complexity of the algorithm for Streett acceptance that goes through Büchi acceptance is exponential, while ours is polynomial. The algorithm for Parity acceptance that goes through Büchi acceptance is in $\boldsymbol{O}(k^3 n^2 m)$ time and $\boldsymbol{O}(k^2 nm)$ space, while ours is in $\boldsymbol{O}(kn^2 m)$ time and $\boldsymbol{O}(nm)$ space, where n (resp. m and k) is the number of control states (resp. transitions and index). Finally, we show that our algorithms yield a better solution for the pushdown model checking problem against linear temporal logic with fairness.

Keywords   pushdown automata, emptiness, ω-words, model checking, linear temporal logic

1   Introduction

Background.   Automata on infinite words were first introduced in the 1960s by Büchi [1] in order to provide a decision procedure for the monadic second-order theory of one successor (i.e., S1S), where the acceptance condition specifies some "good" state that should be visited infinitely often. Muller used automata on infinite words to describe the behavior of non-stabilizing circuits, in which the acceptance condition explicitly specifies all the "good" infinity sets [2]. McNaughton proved a fundamental result in the theory of ω-automata that related different models of automata considered by Büchi and Muller [3], namely, the classes of languages accepted by nondeterministic Büchi automata and by deterministic Muller

---

* Corresponding author (email: songfu@shanghaitech.edu.cn, mzhang@sei.ecnu.edu.cn)

automata are identical. McNaughton introduced a special case of Muller's condition when converting any Büchi automaton into a deterministic Muller automaton which was later formalized by Rabin [4]. The Rabin acceptance condition is a set of pairs of control states, in which a run is termed as accepting if and only if there is a pair such that no "bad" state in the pair is visited infinitely often and some good state in the pair is visited infinitely often. There has been considerable research conducted on the theory of ω-automata. Many variants of acceptance conditions were introduced for ω-automata such as Streett condition [5], Parity condition [6] and Emerson-Lei condition [7] (see survey [8]).

ω-automata play an important role in the automata-theoretic approach to system verification, as a unifying paradigm for the specification, verification, and synthesis of nonterminating systems [9, 10]. The basic idea is that to verify a system against some property, the system is modeled as a finite state automaton, the property is described by an ω-automaton which may be constructed from a Linear temporal logic (LTL) formula [11], and the verification problem is reduced to the emptiness problem of an ω-automaton, which is the "product" of the system automaton and the property ω-automaton. Therefore, considerable effort has already been invested into the theory of ω-automata: in particular, the transformation, expressiveness and emptiness problems. For example, it was proved that the expressiveness of deterministic Büchi automata is weaker than that of nondeterministic Büchi automata, while the later are as expressive as deterministic/nondeterministic ω-automata with Parity, Rabin, Streett or Muller acceptance [12–14]. Deterministic Rabin/Streett/Muller automata are exponentially more succinct than deterministic Parity automata, while nondeterministic Streett automata are exponentially more succinct than nondeterministic Büchi automata. The emptiness problem for nondeterministic Büchi automata (resp. Parity, Rabin, Streett and Muller automata) can be solved in linear time or NLOGSPACE (resp. polynomial time) [15–17].

Pushdown automata over infinite words (called ω-PDAs) are a generation of standard ω-automata [18–22]. The languages recognized by ω-PDAs are called ω-context-free languages. These researches concentrated on the relation between ω-grammars and ω-PDAs. It was independently shown in [18, 21] that ω-PDAs with Muller acceptance can be transformed into one with Büchi acceptance resulting in an exponential blowup in the size of the automaton. ω-PDAs are a very useful model, in particular, for the verification of recursive programs [23–25]. Therefore, efficient algorithms for the emptiness problem of ω-PDAs with Büchi acceptance were developed in [24]. Besides these results, the algorithmic theory of ω-PDAs has not been thoroughly investigated in the literature.

Contribution. In this work, we investigate the emptiness problem of ω-PDAs with respect to the five standard acceptance conditions, namely, Büchi, Parity, Rabin, Streett and Muller acceptances. We use the acronyms in $\{B, P, R, S, M\} \times \{\text{PDA}\}$ to denote the different types of ω-PDAs. The first symbol stands for the type of acceptance condition: B for Büchi, P for Parity, R for Rabin, S for Streett and M for Muller. For example PPDAs denote ω-PDAs with Parity acceptance and SPDAs denote ω-PDAs with Streett acceptance.

The emptiness problem of BPDAs was studied in [23, 24] which was applied to perform LTL model checking on pushdown systems. One can easily transform ω-PDAs with other acceptance types to BPDAs by adapting the transformation for ω-automata (e.g., [13, 14]). In a nutshell, the transformation acts on the control states of ω-PDAs and the stack does not play a role. However, this approach is not optimal. We propose direct and efficient algorithms to the emptiness problems of PPDAs, RPDAs, SPDAs and MPDAs. The results are shown in Table 1. For instance, the best known algorithm for the emptiness problem of BPDAs is in $\boldsymbol{O}(n^2m)$ time and $\boldsymbol{O}(nm)$ space, where n (resp. m) denotes the number of control states (resp. transitions) [24]. The resulting BPDA from an SPDA with n control states, m transitions and k pairs of accepting control states using the best transformation (to the best of our knowledge) has $\boldsymbol{O}(n2^{\min(k,n)})$ control states and $\boldsymbol{O}(m2^{\min(k,n)})$ transitions. Therefore, the best algorithm to the emptiness problem of SPDAs that goes through BPDAs is in $\boldsymbol{O}(k^3n^2m2^{3\min(k,n)})$ time and $\boldsymbol{O}(k^2nm2^{2\min(k,n)})$ space. Our efficient algorithm works in $\boldsymbol{O}(k^3n^2m2^{\min(n,k)})$ time and $\boldsymbol{O}(k^2nm)$ space.

The benefit of using our emptiness checking algorithm is even greater when we consider the model checking problem with fairness. It was shown that the model checking problem for linear temporal logic

Table 1 Complexities of the emptiness problem, where n (resp. m and k) is the number of control states (resp. transitions and index or pairs), $\{F_1, \ldots, F_k\}$ is the Muller acceptance

Acceptance	Büchi	Rabin/Parity	Streett	Muller
Time	$O(n^2m)$	$O(k^3n^2m)$	$O(k^3n^2m2^{3\min(k,n)})$	$O(n^2m(\sum_{i=1}^k|F_i|)^3)$
Space	$O(nm)$	$O(k^2nm)$	$O(k^2nm2^{2\min(k,n)})$	$O(nm(\sum_{i=1}^k|F_i|)^2)$
Our result	–	Theorem 4	Theorem 5	Theorem 6
Time	–	$O(kn^2m)$	$O(k^3n^2m2^{\min(n,k)})$	$O(m \cdot \max(n^2, \sum_{i\in[k]}|F_i|^5))$
Space	–	$O(nm)$	$O(k^2nm)$	$O(m \cdot \max\{n, |F_i|^3 \mid i \in [k]\})$

(LTL) on pushdown systems can be reduced to the emptiness problem of BPDAs [23, 24]. When fairness is considered, it is better to solve the model checking problem by reducing to the emptiness problem of SPDAs rather than BPDAs. In particular, the time complexity is polynomial in the size of the weak fairness when reducing to the emptiness problem of SPDAs, while it is exponential if we use BPDAs.

Related work. ω-PDAs were first proposed and studied by Linna [18–20], Cohen and Gold [21, 22]. Expressiveness of BPDAs and MPDAs were studied in several works such as [18–20]. After about two decades later, the emptiness problem of BPDAs and PPDAs were studied for model checking pushdown systems against linear-time temporal logics such as LTL and linear-time μ-calculus in [23–26]. The eEmptiness problem of BPDAs was independently studied by Bouajjani et al. [23] and Finkel et al. [26] for model checking pushdown systems against LTL. Esparza et al. [24] proposed an efficient implementation algorithm for the emptiness problem of BPDAs which results in a software model checker Moped [25].

Besides the above researches, the emptiness problem of the generalization of ω-PDAs with one letter (also known as, alphabet-free), called alternating ω-PDAs, was investigated for model checking pushdown systems against branching-time temporal logics such as CTL, CTL* and μ-calculus, or equivalent problems. Walukiewicz [27] investigated the pushdown game with Parity objectives and pushdown model checking for μ-calculus which are equivalent to the emptiness problem of alternating PPDAs. Hague and Ong [28, 29] proposed efficient algorithms for winning regions of the pushdown game with Parity objectives and the model checking problem of pushdown systems against μ-calculus respectively which lead to the software model checker PDSolver. Another equivalent instance of this problem, the emptiness problem of pushdown tree automata was studied by Kupferman et al. [30].

Cachat studied the pushdown game with Büchi objectives which is also equivalent to the emptiness problem of alternating BPDAs. Song and Touili [31, 32] proposed an efficient algorithm for the emptiness problem of alternating BPDAs and applied it to solve the model checking problem for CTL on pushdown systems. This algorithm was later implemented in the software model checker PuMoC [33].

The remainder of this paper is structured as follows. Section 2 introduces basic notations, and revisits the results of the transformation of ω-PDAs. In Section 3, we propose efficient algorithms to the emptiness problems of RPDAs, PPDAs, SPDAs and MPDAs. In Section 4, we propose an alternative approach to the model checking problem for LTL on pushdown systems while considering fairness, which is reduced to the emptiness problem of ω-PDAs. Moreover, we discuss the efficiency of the underlying approach. Finally, Section 5 presents conclusion and future work.

2 Preliminaries

We denote by $[k]$ the set $\{1, \ldots, k\}$ for a natural number k. Given a finite alphabet Σ consisting of letters, a finite word over Σ is a sequence $w = a_0 \cdots a_n$ of letters from Σ, while an infinite word over Σ is an infinite sequence $w = a_0 a_1 \cdots$ of letters from Σ. We denote by Σ^* and Σ^ω respectively the sets of all the finite words and infinite words over Σ, ϵ the empty word with $|\epsilon| = 0$. Given a sequence $w = a_0 a_1 \cdots$, we use w_i to denote the letter a_i.

2.1 ω-pushdown automata

Definition 1. An ω-pushdown automaton (ω-PDA for short) is a tuple $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \bot, \mathcal{F})$, where

Σ is the finite input alphabet, P is a finite set of control states, $p_0 \in P$ is the initial control state, Γ is a finite stack alphabet, $\bot \in \Gamma$ is the initial stack content, $\Delta \subseteq (P \times \Gamma) \times \Sigma \times (P \times \Gamma^*)$ is a transition relation such that for every $((p, \bot), a, (p', u)) \in \Delta$, u is in the form of $u'\bot$, and $\mathcal{F}$ is an acceptance condition that will be defined below.

A pushdown system (PDS) $\mathcal{P} = (P, \Gamma, \Delta, p_0, \bot)$ is an ω-PDA if the input alphabet Σ and acceptance condition $\mathcal{F}$ are omitted. The transition relation and runs are defined as for ω-PDAs.

A configuration of the ω-PDA $\mathcal{P}$ is a tuple $\langle p, u \rangle \in P \times \Gamma^*$ and $\langle p_0, \bot \rangle$ is the initial configuration. Let $\mathbb{C}_{\mathcal{P}}$ denote the set $P \times \Gamma^*$. For every $((p, \gamma), a, (p', u)) \in \Delta$, we will use $\langle p, \gamma \rangle \stackrel{a}{\hookrightarrow}_{\mathcal{P}} \langle p', u \rangle$ instead. Given a word $u \in \Gamma^*$, let $|u|$ denote the length of the word u, $|\langle p, u \rangle|$ denote the length $|u|$ of the stack content u, for $p \in P, u \in \Gamma^*$. For every $u' \in \Gamma^*$, if $\langle p, \gamma \rangle \stackrel{a}{\hookrightarrow}_{\mathcal{P}} \langle p', u \rangle$, the configuration $\langle p', uu' \rangle$ is an immediate successor of the configuration $\langle p, \gamma u' \rangle$, denoted by, $\langle p, \gamma u' \rangle \stackrel{a}{\Longrightarrow}_{\mathcal{P}} \langle p', uu' \rangle$. Intuitively, if $\langle p, \gamma \rangle \stackrel{a}{\hookrightarrow}_{\mathcal{P}} \langle p', u \rangle$, then the ω-PDA $\mathcal{P}$ can move into the configuration $\langle p', uu' \rangle$ when it is at the configuration $\langle p, \gamma u' \rangle$ and reads the input letter a. A pair $\langle p, \gamma \rangle$ such that $p \in P$ and $\gamma \in \Gamma$ is called head. We also abbreviate $\mathbb{C}_{\mathcal{P}}$, $\hookrightarrow_{\mathcal{P}}$ and $\Longrightarrow_{\mathcal{P}}$ as $\mathbb{C}$, $\hookrightarrow$ and $\Longrightarrow$ if $\mathcal{P}$ is clear from the context.

Let $\Longrightarrow^+, \Longrightarrow^* \subseteq \mathbb{C} \times \Sigma^* \times \mathbb{C}$ be the smallest relations such that the following conditions hold:

- $c \stackrel{\epsilon}{\Longrightarrow}^* c$ for every $c \in \mathbb{C}$,
- $c \stackrel{aw}{\Longrightarrow}^* c''$ and $c \stackrel{aw}{\Longrightarrow}^+ c''$, if $c \stackrel{a}{\Longrightarrow} c'$ and $c' \stackrel{w}{\Longrightarrow}^* c''$.

Intuitively, $c \stackrel{w}{\Longrightarrow}^* c'$ denotes that the ω-PDA $\mathcal{P}$ can move into the configuration c' when $\mathcal{P}$ is at the configuration c and reads the input word w.

Let $\mathrm{pre}^* : 2^{\mathbb{C}} \to 2^{\mathbb{C}}$ be a function such that $\mathrm{pre}^*(C) = \{c \in \mathbb{C} \mid \exists w \in \Sigma_\epsilon^*, \exists c' \in C : c \stackrel{w}{\Longrightarrow}^* c'\}$ for every $C \subseteq \mathbb{C}$. Given a set of configurations $C \subseteq \mathbb{C}_{\mathcal{P}}$ and a configuration $c \in \mathbb{C}$, the reachability problem is to determine whether $c \in \mathrm{pre}^*(C)$ or not. Given a control state $p \in P$ and a configuration $c \in \mathbb{C}$, the control state reachability problem is to determine whether $c \in \mathrm{pre}^*(\{p\} \times \Gamma^*)$.

Given an infinite word $w = a_0 a_1 \cdots \in \Sigma^\omega$, a run of the ω-PDA $\mathcal{P}$ on w is an infinite sequence $\rho = c_0 c_1 \cdots$ of configurations such that $c_0 = \langle p_0, \bot \rangle$, for every $i \geqslant 0$, there is $a_i' \in \Sigma$ with $c_i \stackrel{a_i'}{\Longrightarrow} c_{i+1}$, and $w = a_0' a_1' \cdots$. A path of the ω-PDA $\mathcal{P}$ is a finite sequence of configurations which is a prefix of a run. A path starting from a configuration c and ending with a configuration c' over a word w is sometimes abbreviated as $c \stackrel{w}{\Longrightarrow}^* c'$. Given a run ρ, let $\mathsf{Inf}(\rho)$ denote the set of control states visited infinitely often in the run ρ. In this work, we consider the following five well-known acceptance conditions in the literature.

- Büchi acceptance [1]: $\mathcal{F} \subseteq P$ is a finite set of accepting control states. A run ρ is accepting if and only if $\mathsf{Inf}(\rho) \cap \mathcal{F} \neq \emptyset$.

- Parity acceptance [6]: $\mathcal{F} : P \to [k]$ is a function that assigns a priority to each control state, where k is some natural number called index. A run ρ is accepting if and only if $\min(\{\mathcal{F}(p) \mid p \in \mathsf{Inf}(\rho)\})$ is even.

- Rabin acceptance [4]: $\mathcal{F} = \{(E_1, F_1), \ldots, (E_k, F_k)\}$ is a set of pairs of control states with $E_i, F_i \subseteq P$ for every $i \in [k]$. A run ρ is accepting if and only if there exists some $i \in [k]$ such that $E_i \cap \mathsf{Inf}(\rho) = \emptyset$ and $F_i \cap \mathsf{Inf}(\rho) \neq \emptyset$.

- Streett acceptance [5]: $\mathcal{F} = \{(E_1, F_1), \ldots, (E_k, F_k)\}$ is a set of pairs of control states with $E_i, F_i \subseteq P$ for every $i \in [k]$. A run ρ is accepting if and only if for all $i \in [k]$, $E_i \cap \mathsf{Inf}(\rho) \neq \emptyset$ or $F_i \cap \mathsf{Inf}(\rho) = \emptyset$.

- Muller acceptance [2]: $\mathcal{F} = \{F_1, \ldots, F_k\}$ is a set of sets of control states with $F_i \subseteq P$ for every $i \in [k]$. A run ρ is accepting if and only if $\mathsf{Inf}(\rho) \in \mathcal{F}$.

The ω-PDA $\mathcal{P}$ accepts an infinite word $w \in \Sigma^\omega$ if and only if it has an accepting run over the input word w. Let $\mathcal{L}(\mathcal{P}) \subseteq \Sigma^\omega$ denote the set of infinite words accepted by the ω-PDA $\mathcal{P}$, called the language of $\mathcal{P}$. An ω-PDA $\mathcal{P}$ is empty if and only if $\mathcal{L}(\mathcal{P}) = \emptyset$. Two ω-PDAs $\mathcal{P}_1$ and $\mathcal{P}_2$ are equivalent if $\mathcal{L}(\mathcal{P}_1) = \mathcal{L}(\mathcal{P}_2)$. Given an ω-PDA $\mathcal{P}$, its emptiness problem is to determine whether $\mathcal{L}(\mathcal{P}) = \emptyset$ or not. W.l.o.g., we assume that the input alphabet is singleton. Indeed, for the emptiness problem, the finite alphabet can be abstracted as a singleton set.

We will use $\mathcal{P}$-automata as "data structures" to finitely represent infinite sets of configurations.

Table 2 Summary of the transformation for ω-PDAs, where n (resp. m and k) is the number of control states (resp. transitions and index/pairs), $\sharp$Ctrl. (resp. $\sharp$Trans.) denotes the number of control states (resp. transition rules) in the resulting ω-PDAs, and $\|\mathcal{F}\| = \sum_{i=1}^{k} |F_i|$

Acceptance	R2B	P2B	S2B	M2B
$\sharp$Ctrl.	$\boldsymbol{O}(kn)$	$\boldsymbol{O}(kn)$	$\boldsymbol{O}(kn2^{\min(k,n)})$	$\boldsymbol{O}(n\|\mathcal{F}\|)$
$\sharp$Trans.	$\boldsymbol{O}(km)$	$\boldsymbol{O}(km)$	$\boldsymbol{O}(km2^{\min(k,n)})$	$\boldsymbol{O}(m\|\mathcal{F}\|)$

2.2 $\mathcal{P}$-automata

Definition 2 ([23]). Given an ω-PDA $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \bot, \mathcal{F})$, a $\mathcal{P}$-automaton is a tuple $\mathcal{A} = (Q, \Gamma, \delta, Q_0, F)$, where Q is a finite set of control states with $P \subseteq Q$, $Q_0 \subseteq Q$ is a finite set of initial states, $F \subseteq Q$ is a finite set of final states and $\delta \subseteq Q \times \Gamma \times Q$ is a transition relation.

We write $p \xrightarrow{\gamma} q$ instead of $(p, \gamma, q) \in \delta$. We define the transition relation $\longrightarrow^* \subseteq Q \times \Gamma^* \times Q$ as the smallest relation such that

- $q \xrightarrow{\epsilon}^* q$, for every $q \in Q$,
- $q \xrightarrow{\gamma u}^* q''$, if $q \xrightarrow{\gamma} q'$ and $q' \xrightarrow{u}^* q''$.

A configuration $\langle p, u \rangle \in \mathbb{C}$ is recognized (accepted) by $\mathcal{A}$ if $p \xrightarrow{u}^* q$ such that $p \in Q_0$ and $q \in F$. Let $\mathcal{L}(\mathcal{A})$ be the set of configurations recognized by $\mathcal{A}$. A set of configurations $C \subseteq \mathbb{C}$ is regular if there is a $\mathcal{P}$-automaton $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) = C$. Given a configuration $\langle p, u \rangle$, whether $\langle p, u \rangle$ is accepted by $\mathcal{A}$ can be ecided in polynomial time of the size of $\mathcal{A}$ and $|u|$.

Theorem 1 ([24]). Given an ω-PDA $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \bot, \mathcal{F})$ and a regular set of configurations $C \subseteq \mathbb{C}$ recognized by a $\mathcal{P}$-automaton $\mathcal{A} = (Q, \Gamma, \delta, Q_0, F)$, a new $\mathcal{P}$-automaton $\mathcal{A}'$ can be constructed in $\boldsymbol{O}(|Q|^2 \cdot |\Delta|)$ time and $\boldsymbol{O}(|Q| \cdot |\Delta| + |\delta|)$ space such that $\mathcal{L}(\mathcal{A}') = \text{pre}^*(C)$.

The emptiness problem of BPDAs was shown in PTIME by Bouajjani et al. [23]. Later, Esparza et al. [24] proposed an efficient algorithm for this problem.

Theorem 2 ([24]). The emptiness problem of BPDAs $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \bot, \mathcal{F})$ can be decided in $\boldsymbol{O}(|P|^2 \cdot |\Delta|)$ time and $\boldsymbol{O}(|P| \cdot |\Delta|)$ space.

By leveraging the well-known transformation for ω-automata (e.g., [13, 14]), we can obtain the results shown in Table 2. In a nutshell, the transformation acts on the control states of ω-PDAs and the stack does not play a role. By applying Theorem 2, we can obtain Theorem 3.

Theorem 3. The emptiness problems of ω-PDAs $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \bot, \mathcal{F})$ (with k index/pairs in $\mathcal{F}$) can be solved in

- $\boldsymbol{O}(k^3|P|^2 \cdot |\Delta|)$ time and $\boldsymbol{O}(k^2|P| \cdot |\Delta|)$ space for RPDAs and PPDAs.
- $\boldsymbol{O}(k^3|P|^2 \cdot |\Delta| \cdot 2^{3\min(k,|P|)})$ time and $\boldsymbol{O}(k^2|P| \cdot |\Delta| \cdot 2^{2\min(k,|P|)})$ space for SPDAs.
- $\boldsymbol{O}(|P|^2 \cdot |\Delta| \cdot \sum_{i=1}^{k} |F_i|^3)$ time and $\boldsymbol{O}(|P| \cdot |\Delta| \cdot \sum_{i=1}^{k} |F_i|^2)$ space for MPDAs.

Proof. For self-contained, we present the transformations here. Given an RPDA $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \bot, \mathcal{F})$ with $\mathcal{F} = \{(E_1, F_1), \ldots, (E_k, F_k)\}$, one can construct an equivalent BPDA $\mathcal{P}' = (\Sigma, P', \Gamma, \Delta', p_0, \bot, \mathcal{F}')$ such that $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{P}')$, where $|P'| = \boldsymbol{O}(k|P|)$ and $|\Delta'| = \boldsymbol{O}(k|\Delta|)$.

$\mathcal{P}'$ consists of $k + 1$ copies of $\mathcal{P}$ in which one copy fully behaves the same as $\mathcal{P}$. The other k copies are partial and each of them is associated to a pair (E_i, F_i), its control states are labeled by i and it contains the control states from $P \setminus E_i$. A run of $\mathcal{P}'$ starting from the initial configuration $\langle p_0, \bot \rangle$ initially runs at the full copy. During the run, $\mathcal{P}'$ can nondeterministically choose between staying in the full copy or moving to one of the other k copies that guesses a proper pair (E_i, F_i) in order to have an accepting run in $\mathcal{P}$. Once, $\mathcal{P}'$ enters a copy associated to a pair (E_i, F_i), the run does not go into any configurations with control states in E_i. The Büchi acceptance $\mathcal{F}' = \bigcup_{i=1}^{k} F_i \times \{i\}$ ensures that the run of $\mathcal{P}'$ will eventually leave the full copy and move to some configurations labeled by i. Subsequently, it will visit some configurations with control states in F_i infinitely often.

Formally, for every $i \in [k]$, let $P_i = (P \setminus E_i) \times \{i\}$. We define $P' = P \cup \bigcup_{i=1}^{k} P_i$, $\mathcal{F}' = \bigcup_{i=1}^{k} F_i \times \{i\}$, Δ' is the smallest transition relation such that the following conditions hold:

- For every $\langle p, \gamma \rangle \xrightarrow{a} \langle p', u \rangle \in \Delta$, $\langle p, \gamma \rangle \xrightarrow{a} \langle p', u \rangle \in \Delta'$ and for every $i \in [k]$ with $p' \in P \setminus E_i$, $\langle p, \gamma \rangle \xrightarrow{a} \langle (p', i), u \rangle \in \Delta'$;

- For every $i \in [k]$ and $\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', u \rangle \in \Delta$ such that $p, p' \in P \setminus E_i$, $\langle (p, i), \gamma \rangle \overset{a}{\hookrightarrow} \langle (p', i), u \rangle \in \Delta'$.

It is easy to verify that $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{P}')$. Given an PPDA $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \bot, \mathcal{F})$ with $\mathcal{F} : P \to [k]$, one can construct an equivalent BPDA $\mathcal{P}' = (\Sigma, P', \Gamma, \Delta', p_0, \bot, \mathcal{F}')$ such that $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{P}')$, $|P'| = \boldsymbol{O}(k|P|)$ and $|\Delta'| = \boldsymbol{O}(k|\Delta|)$.

Without loss of generality, we assume that $k = 2m + 1$ for some natural number m. For the Parity acceptance $\mathcal{F} : P \to [k]$, we can construct the Rabin chain $\mathcal{F}' = \{(E_1, F_1), \ldots, (E_m, F_m)\}$ such that for every $i \in [m]$,

$$E_i = \{p \in P \mid \mathcal{F}(p) \leqslant 2i - 1\} \text{ and } F_i = \{p \in P \mid \mathcal{F}(p) \leqslant 2i\}.$$

Let $\mathcal{P}' = (\Sigma, P, \Gamma, \Delta, p_0, \bot, \mathcal{F}')$ be the RPDA. Hence, $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{P}')$. We can construct an BPDA $\mathcal{P}'' = (\Sigma, P', \Gamma, \Delta', p_0, \bot, \mathcal{F}'')$ such that $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{P}'')$, where $|P'| = \boldsymbol{O}(k|P|)$ and $|\Delta'| = \boldsymbol{O}(k|\Delta|)$. Given an SPDA $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \bot, \mathcal{F})$ with $\mathcal{F} = \{(E_1, F_1), \ldots, (E_k, F_k)\}$, we can construct an equivalent BPDA $\mathcal{P}' = (\Sigma, P', \Gamma, \Delta', p_0, \bot, \mathcal{F}')$ such that $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{P}')$, where $|P'| = \boldsymbol{O}(k|P| \cdot 2^{\min(k, |P|)})$ and $|\Delta'| = \boldsymbol{O}(k|\Delta| \cdot 2^{\min(k, |P|)})$.

We guess a subset $S \subseteq P$ of control states such that only control states in S are visited after some point. To meet the Streett acceptance, we only need to ensure that for every $i \in [k]$, if some control state in S appears in F_i, then some control state from E_i should be infinitely often visited. Intuitively, suppose that the control state $p \in S \cap F_i$, then some control state in E_i should be visited infinitely often, irrespective of whether p is visited infinitely often or not. This acceptance is stronger than the Streett acceptance and can be regarded as the generalized Büchi acceptance which can be transformed into Büchi acceptance via de-generalization.

Given a set $S \subseteq P$, let $\mathsf{In}(S) = \{E_i \mid i \in [k], F_i \cap S \neq \emptyset\}$. Let $\mathsf{In}(\mathcal{P}) \subseteq 2^P$ be a set such that the following conditions hold:

- If there exists some set $S \subseteq P$ with some set $\mathsf{In}(S)$, then there exists a set $S' \in \mathsf{In}(\mathcal{P})$ such that $\mathsf{In}(S) = \mathsf{In}(S')$ and $S \subseteq S'$,

- For every two distinct sets $S, S' \in \mathsf{In}(\mathcal{P})$, $\mathsf{In}(S) \neq \mathsf{In}(S')$.

It is easy to see that $|\mathsf{In}(\mathcal{P})|$ is at most $2^{\min(k, |P|)}$. Suppose that $\mathsf{In}(\mathcal{P}) = \{S_1, \ldots, S_n\} \subseteq 2^P$ and elements in $\mathsf{In}(S)$ are well-ordered, we use $\mathsf{In}(S, i)$ to denote the ith element. We define $P' = P \cup P \times \{S_1, \ldots, S_n\} \times \{0, 1, \ldots, k\}$, $\mathcal{F}' = \{(p, S_i, |\mathsf{In}(S_i)|) \mid p \in S_i, i \in [n]\}$, Δ' is the smallest transition relation such that the following conditions hold:

(1) For every $\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', u \rangle \in \Delta$,

- $\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', u \rangle \in \Delta'$ and
- $\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle (p', S_i, 0), u \rangle \in \Delta'$ for every $i \in [n]$ with $p' \in S_i$;

(2) For every $i \in [n]$, every $\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', u \rangle \in \Delta$ with $p, p' \in S_i$ and every $j \in \{0, \ldots, |\mathsf{In}(S_i)| - 1\}$,

- $\langle (p, S_i, j), \gamma \rangle \overset{a}{\hookrightarrow} \langle (p', S_i, j + 1), u \rangle \in \Delta'$, if $p' \in \mathsf{In}(S_i, j + 1)$,
- $\langle (p, S_i, j), \gamma \rangle \overset{a}{\hookrightarrow} \langle (p', S_i, j), u \rangle \in \Delta'$, if $p' \in P \setminus \mathsf{In}(S_i, j + 1)$,
- $\langle (p, S_i, |\mathsf{In}(S_i)|), \gamma \rangle \overset{a}{\hookrightarrow} \langle (p', S_i, 0), u \rangle \in \Delta'$.

Initially, $\mathcal{P}'$ runs the same as $\mathcal{P}$. $\mathcal{P}'$ nondeterministically guesses a set S_i stored into the control states with a counter starting from 0. The set S_i consists of control states that are potentially visited infinitely. Once S_i is selected, the runs of $\mathcal{P}'$ can only visit control states from S_i. For each configuration with the counter j, the counter will add 1 if a control state from $\mathsf{In}(S_i, j + 1)$ is visited and the counter will be reset to 0 if j is equal to $|\mathsf{In}(S_i)|$. At least one control state in each set of $\mathsf{In}(S)$ is infinitely often visited if and only if the counter is set to $|\mathsf{In}(S)|$ infinitely often. This implies that for each set E_j, if some control state p from E_j is visited infinitely often, then p must be in S_i. Therefore, some control state in F_j will be visited infinitely often owing to the condition that the counter is set to $|\mathsf{In}(S)|$ infinitely often. Now, we obtain that $\mathcal{L}(\mathcal{P}') \subseteq \mathcal{L}(\mathcal{P})$. $\mathcal{L}(\mathcal{P}) \subseteq \mathcal{L}(\mathcal{P}')$ is also true, because given an accepting run of $\mathcal{P}$, such a set S_i always exists.

Given an MPDA $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \bot, \mathcal{F})$ with $\mathcal{F} = \{F_1, \ldots, F_k\}$, one can construct an equivalent BPDA $\mathcal{P}' = (\Sigma, P', \Gamma, \Delta', p_0, \bot, \mathcal{F}')$ such that $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{P}')$, where $|P'| = \boldsymbol{O}(|P| \cdot \|\mathcal{F}\|)$ and $|\Delta'| = \boldsymbol{O}(|\Delta| \cdot \|\mathcal{F}\|)$.

For every $i \in [k]$, suppose that $F_i = \{p_i^1, \ldots, p_i^{|F_i|}\}$. $\mathcal{P}'$ guesses the set F_i and the position in the accepting run of $\mathcal{P}$ from which only configurations with control states in F_i are visited. Runs of $\mathcal{P}'$ starting from the initial configuration $\langle p_0, \perp \rangle$ run as in $\mathcal{P}$. The runs can nondeterministically choose between staying in the copy of $\mathcal{P}$ or moving to one of the other copies labeled by i which guess a proper set F_i. When moving to a copy labeled by i, the runs set a counter j to 0 and increase the counter by 1 when the control state p_i^{j+1} is visited. The counter j will be reset to 0 once j is $|F_i|$ meaning that all the control states from F_i are visited. If the counter is infinitely often to $|F_i|$, then the Muller acceptance is met.

Formally, $P' = P \cup \{(p, j, i) \mid p \in F_i, 0 \leqslant j \leqslant |F_i|, i \in [k]\}$, $\mathcal{F}' = \{(p, |F_i|, i) \mid p \in F_i, i \in [k]\}$ and Δ' is the smallest transition relation such that the following conditions hold: For every $\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', u \rangle \in \Delta$,

- $\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', u \rangle \in \Delta'$;
- $\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle (p', 0, i), u \rangle \in \Delta'$ for every $i \in [k]$ such that $p' \in F_i$;
- $\langle (p, j, i), \gamma \rangle \overset{a}{\hookrightarrow} \langle (p', j+1, i), u \rangle \in \Delta'$, if $p \in F_i$ and $p' = p_i^{j+1}$;
- $\langle (p, j, i), \gamma \rangle \overset{a}{\hookrightarrow} \langle (p', j, i), u \rangle \in \Delta'$, if $p, p' \in F_i, p' \neq p_i^{j+1}$ and $j < |F_i|$;
- $\langle (p, |F_i|, i), \gamma \rangle \overset{a}{\hookrightarrow} \langle (p', 0, i), u \rangle \in \Delta'$, if $p, p' \in F_i$.

We can see that $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{P}')$.

However, the approach going through BPDAs is not optimal. In the rest of this paper, we propose more efficient algorithms.

3 Efficient solution to the emptiness problem

In this section, we propose efficient solutions to the emptiness problems of RPDAs, PPDAs, SPDAs and MPDAs and discuss their lower bounds. For this purpose, we first recall the concept of repeating heads and some results for the emptiness problem of BPDAs from [24].

3.1 Repeating heads

Given an BPDA $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \perp, \mathcal{F})$, let $\Longrightarrow^{\mathsf{B}} \subseteq \mathbb{C} \times \Sigma^* \times \mathbb{C}$ be the relation defined as follows: $c \overset{w}{\Longrightarrow}^{\mathsf{B}} c'$ if and only if $c \overset{w_1}{\Longrightarrow}^* c_f \overset{w_2}{\Longrightarrow}^+ c'$, for some $c_f \in \mathcal{F} \times \Gamma^*$ and $w = w_1 w_2$. Intuitively, $c \overset{w}{\Longrightarrow}^{\mathsf{B}} c'$ denotes that the BPDA $\mathcal{P}$ can move into the configuration c' when $\mathcal{P}$ is at the configuration c and reads the input word w. Furthermore, a configuration whose first component is an accepting control state is visited.

A head $\langle p, \gamma \rangle \in P \times \Gamma$ is repeating if there exist some $u \in \Gamma^*, w \in \Sigma^*$ such that $\langle p, \gamma \rangle \overset{w}{\Longrightarrow}^{\mathsf{B}} \langle p, \gamma u \rangle$. Let $\boldsymbol{R}_{\mathsf{B}}$ be the set of repeating heads in $\mathcal{P}$. Let $\boldsymbol{R}_{\mathsf{B}} \Gamma^*$ denote the set of configurations $\{\langle p, \gamma u \rangle \in \mathbb{C} \mid \langle p, \gamma \rangle \in \boldsymbol{R}_{\mathsf{B}}, u \in \Gamma^* \}$.

Proposition 1 ([23, 24]). The BPDA $\mathcal{P}$ is empty if and only if $\langle p_0, \perp \rangle \notin \mathrm{pre}^*(\boldsymbol{R}_{\mathsf{B}} \Gamma^*)$.

To compute the repeating heads, Esparza et al. proposed a polynomial-time algorithm which is an adaption of the pre* algorithm.

Lemma 1 ([24]). The set $\boldsymbol{R}_{\mathsf{B}}$ of repeating heads of an BPDA $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \perp, \mathcal{F})$ can be computed in $\boldsymbol{O}(|P|^2 \cdot |\Delta|)$ time and $\boldsymbol{O}(|P| \cdot |\Delta|)$ space.

3.2 Efficient solution to the emptiness problem of RPDAs

We fix the RPDA $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \perp, \mathcal{F})$ with $\mathcal{F} = \{(E_1, F_1), \ldots, (E_k, F_k)\}$. Our approach is based on the observation that for every accepting run ρ, there is a position $m \geqslant 0$ and an indicator $i \in [k]$ such that for every $n \geqslant m$, the configuration $\rho(n)$ is from $(P \setminus E_i) \times \Gamma^*$, namely, the control states from E_i do not appear after the position m. In addition, some control states from F_i appear infinitely often in ρ after the position m. Therefore, the problem can be reduced to compute heads $\langle p, \gamma \rangle$ the form of $\langle p, \gamma \rangle \overset{w}{\Longrightarrow}^+ \langle p, \gamma u \rangle$ such that there exists the indicator $i \in [k]$ with no control state from E_i appearing in

$\langle p, \gamma \rangle \overset{w}{\Longrightarrow}^+ \langle p, \gamma u \rangle$, but some control states from F_i appearing in $\langle p, \gamma \rangle \overset{w}{\Longrightarrow}^+ \langle p, \gamma u \rangle$. We show that such kind of heads can be computed by applying Lemma 1.

For every $i \in [k]$, let $\overset{*}{\Longrightarrow_{\overline{i}}}, \overset{+}{\Longrightarrow_{\overline{i}}} \subseteq \mathbb{C} \times \Sigma^* \times \mathbb{C}$ be the smallest relations such that the following conditions hold:

- $\langle p, u \rangle \overset{\epsilon}{\Longrightarrow_{\overline{i}}^*} \langle p, u \rangle$, for every control state $p \in P \setminus E_i$ and $u \in \Gamma^*$;
- $\langle p, u \rangle \overset{aw}{\Longrightarrow_{\overline{i}}^*} \langle p'', u'' \rangle$ and $\langle p, u \rangle \overset{aw}{\Longrightarrow_{\overline{i}}^+} \langle p'', u'' \rangle$, if $\langle p, u \rangle \overset{a}{\Longrightarrow} \langle p', u' \rangle$ with $p, p' \in P \setminus E_i$ and $\langle p', u' \rangle \overset{w}{\Longrightarrow_{\overline{i}}^*} \langle p'', u'' \rangle$.

Intuitively, $\langle p, u \rangle \overset{w}{\Longrightarrow_{\overline{i}}^*} \langle p', u' \rangle$ indicates that the PPDA $\mathcal{P}$ can move into the configuration $\langle p', u' \rangle$ when $\mathcal{P}$ is at the configuration $\langle p, u \rangle$ and reads the input word w. Moreover, no control state from E_i appears in the path $\langle p, u \rangle \overset{w}{\Longrightarrow_{\overline{i}}^*} \langle p', u' \rangle$.

For every $i \in [k]$, a head $\langle p, \gamma \rangle \in P \times \Gamma$ is $\overline{i}$-repeating if $\langle p, \gamma \rangle \overset{w_1}{\Longrightarrow_{\overline{i}}^*} \langle p_f, u \rangle \overset{w_2}{\Longrightarrow_{\overline{i}}^+} \langle p, \gamma v \rangle$ for some configuration $\langle p_f, u \rangle \in F_i \times \Gamma^*$. For every $i \in [k]$, let $\boldsymbol{R}_{\mathsf{R}}^i$ be the set of all the $\overline{i}$-repeating heads in $\mathcal{P}$.

Proposition 2. Let $\boldsymbol{R}_{\mathsf{R}}$ denote the set $\bigcup_{i=1}^k \boldsymbol{R}_{\mathsf{R}}^i$, $\mathcal{P}$ is empty if and only if $\langle p_0, \bot \rangle \notin \mathrm{pre}^*(\boldsymbol{R}_{\mathsf{R}} \Gamma^*)$.

Proof. ($\Rightarrow$) Suppose that $\mathcal{P}$ has an accepting run $\rho = c_0 c_1 \cdots$ on the infinite word $w = a_0 a_1 \cdots$, where $c_0 = \langle p_0, \bot \rangle$. We can construct a subsequence $c_{n_1} c_{n_2} \cdots$ of ρ such that

$$|c_{n_1}| = \min\{|c_j| \mid j \geqslant 0\},$$
$$|c_{n_i}| = \min\{|c_j| \mid j > n_{i-1}\}, \ \forall i \geqslant 2.$$

We can see that the sequence $|c_{n_1}|, |c_{n_2}|, \ldots$, strictly increases, implying that for every $i \geqslant 1$, once the configuration c_{n_i} is reached, the rest of the run ρ from the position n_i will never change the stack content in c_{n_i} except for the topmost of the stack. Because the sets P and Γ are finite (hence the number of heads is finite), there must exist a pair $\langle p, \gamma \rangle$ that appears in $c_{n_1} c_{n_2} \cdots$ infinitely often. Therefore, we can construct a subsequence $c_{j_1} c_{j_2} \cdots$ of $c_{n_1} c_{n_2} \cdots$ such that for every $i \geqslant 1$, the head of c_{j_i} is $\langle p, \gamma \rangle$.

Because $\rho = c_0 c_1 \cdots$ is an accepting run, let (E_m, F_m) be a pair such that $\mathsf{Inf}(\rho) \cap F_m \neq \emptyset$ and $\mathsf{Inf}(\rho) \cap E_m = \emptyset$. By the above construction, there exist a position $j_e \geqslant j_1$ and a configuration $\langle p_f, u u_1' \rangle \in F_m \times \Gamma^*$ such that

$$c_{j_e} = \langle p, \gamma u_1 \rangle \overset{w_1}{\Longrightarrow_m^*} \langle p_f, u u_1' \rangle \overset{w_2}{\Longrightarrow_m^+} \langle p, \gamma u_2 u_1'' \rangle = c_{j_{e+1}}.$$

Since once the configuration c_{j_e} is reached, the rest of the run ρ from the position j_e will never change the stack content in c_{j_e} except for the topmost of the stack, we obtain that $u_1 = u_1' = u_1''$ and

$$\langle p, \gamma \rangle \overset{w_1}{\Longrightarrow_m^*} \langle p_f, u \rangle \overset{w_2}{\Longrightarrow_m^+} \langle p, \gamma u_2 \rangle.$$

Therefore, $\langle p, \gamma \rangle \in \boldsymbol{R}_{\mathsf{R}}$. The result immediately follows.

($\Leftarrow$) Suppose that $\langle p_0, \bot \rangle \in \mathrm{pre}^*(\boldsymbol{R}_{\mathsf{R}} \Gamma^*)$. Let $\langle p, \gamma \rangle \in \boldsymbol{R}_{\mathsf{R}}$ be the head such that

$$\langle p_0, \bot \rangle \overset{w_0}{\Longrightarrow^*} \langle p, \gamma u \rangle \text{ and } \langle p, \gamma \rangle \overset{w_1}{\Longrightarrow_{\overline{i}}^*} \langle p_f, u' \rangle \overset{w_2}{\Longrightarrow_{\overline{i}}^+} \langle p, \gamma u'' \rangle$$

for some $i \in [k]$, $p_f \in F_i$. Therefore, we obtain the following accepting run:

$$\langle p_0, \bot \rangle \overset{w_0}{\Longrightarrow^*} \langle p, \gamma u \rangle \overset{w_1}{\Longrightarrow^*} \langle p_f, u' u \rangle \overset{w_2}{\Longrightarrow^+} \langle p, \gamma u'' u \rangle \overset{w_1}{\Longrightarrow^*} \langle p_f, u' u'' u \rangle \overset{w_2}{\Longrightarrow^+} \langle p, \gamma u'' u'' u \rangle \cdots.$$

Note that no control state from E_i is visited in the path $\langle p, \gamma v u \rangle \overset{w_1}{\Longrightarrow^*} \langle p_f, u' v u \rangle \overset{w_2}{\Longrightarrow^+} \langle p, \gamma u'' v u \rangle$ for every $v \in \{u''^i \mid i \geqslant 0\}$.

For every $i \in [k]$, let $\mathcal{P}_{\overline{i}} = (\Sigma, P_{\overline{i}}, \Gamma, \Delta_{\overline{i}}, p_0, \bot, F_i)$ be the BPDA, where $P_{\overline{i}} = P \setminus E_i$, $\Delta_{\overline{i}} = \Delta \cap (P_{\overline{i}} \times \Gamma) \times \Sigma \times (P_{\overline{i}} \times \Gamma^*)$, then, $\mathcal{P}_{\overline{i}}$ does not use any control state from E_i and we obtain Lemma 2.

Lemma 2. For each $i \in [k]$, let $\boldsymbol{R}_{\mathsf{B}}^{\overline{i}}$ be the set of repeating heads in the BPDA $\mathcal{P}_{\overline{i}}$, $\boldsymbol{R}_{\mathsf{B}}^{\overline{i}} = \boldsymbol{R}_{\mathsf{R}}^i$.

Applying Lemma 1, for every $i \in [k]$, $\boldsymbol{R}_{\mathsf{R}}^i$ can be computed in $\boldsymbol{O}(|P|^2 \cdot |\Delta|)$ time and $\boldsymbol{O}(|P| \cdot |\Delta|)$ space. This implies that $\boldsymbol{R}_{\mathsf{R}} = \bigcup_{i=1}^k \boldsymbol{R}_{\mathsf{R}}^i$ can be computed in $\boldsymbol{O}(k|P|^2 \cdot |\Delta|)$ time and $\boldsymbol{O}(|P| \cdot |\Delta|)$ space.

Applying Theorem 1, we can compute a $\mathcal{P}$-automaton exactly accepting the set $\text{pre}^*(\boldsymbol{R}_\mathsf{R}\Gamma^*)$. Therefore, for a given configuration c, whether the RPDA has an accepting run starting from c can be decided by checking whether the $\mathcal{P}$-automaton accepts c or not. Then, we obtain Theorem 4.

Theorem 4. The emptiness problem of RPDAs $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \bot, \mathcal{F})$ with $\mathcal{F} = \{(E_1, F_1), \ldots, (E_k, F_k)\}$ can be decided in $\boldsymbol{O}(k|P|^2 \cdot |\Delta|)$ time and $\boldsymbol{O}(|P| \cdot |\Delta|)$ space.

Notice that $\boldsymbol{R}_\mathsf{R}$ is computed by computing $\boldsymbol{R}_\mathsf{R}^i$ for $i \in [k]$. Because the size of $\boldsymbol{R}_\mathsf{R}$ does not depend on k and the space complexity for computing $\boldsymbol{R}_\mathsf{R}^i$ also does not depend on k, the space complexity of the emptiness problem of RPDAs does not depend on k.

3.3 Efficient solution to the emptiness problem of SPDAs

Let us fix an SPDA $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \bot, \mathcal{F})$ with $\mathcal{F} = \{(E_1, F_1), \ldots, (E_k, F_k)\}$. We propose an efficient solution to the emptiness problem of SPDAs by guessing the set of control states that can be visited after some point. Similar to the transformation from Streett automata to Büchi automata, for every set $S \in \mathsf{In}(\mathcal{P})$, we compute a head $\langle p, \gamma \rangle$ of the form $\langle p, \gamma \rangle \overset{w}{\Longrightarrow}^+ \langle p, \gamma u \rangle$. Furthermore, only control states from S are visited in $\langle p, \gamma \rangle \overset{w}{\Longrightarrow}^+ \langle p, \gamma u \rangle$ and for every $i \in [k]$, if $F_i \cap S \neq \emptyset$, at least one control state from E_i is visited in $\langle p, \gamma \rangle \overset{w}{\Longrightarrow}^+ \langle p, \gamma u \rangle$. These heads can be computed by applying Lemma 1 on some proper BPDAs. Then, by applying pre^* computation to these heads, we can compute an $\mathcal{P}$-automaton $\mathcal{A}$ recognizing the set of configurations from which the SPDA has an accepting run.

Given a set $S \subseteq P$, let $R^S \subseteq P$ be the repeating heads of the BPDA $\mathcal{P}_S = (\Sigma, P', \Gamma, \Delta', (p_0, 0), \bot, \mathcal{F}')$, where $\mathcal{F}' = S \times \{|\mathsf{In}(S)|\}$, P' and Δ' are defined as follows: for every $\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', u' \rangle \in \Delta$ with $p, p' \in S$,

- $\langle (p, i), \gamma \rangle \overset{a}{\hookrightarrow} \langle (p', i+1), u \rangle \in \Delta'$, for every $i \in \{0, \ldots, |\mathsf{In}(S)| - 1\}$ with $p' \in \mathsf{In}(S, i+1)$;
- $\langle (p, i), \gamma \rangle \overset{a}{\hookrightarrow} \langle (p', i), u \rangle \in \Delta'$, for $i \in \{0, \ldots, |\mathsf{In}(S)| - 1\}$;
- $\langle (p, |\mathsf{In}(S)|), \gamma \rangle \overset{a}{\hookrightarrow} \langle (p', 0), u \rangle \in \Delta'$.

Proposition 3. Let $\boldsymbol{R}_\mathsf{S} = \{\langle p, \gamma \rangle \in P \times \Gamma \mid \exists S' \subseteq P : \langle (p, 0), \gamma \rangle \in R^{S'}\}$, the SPDA $\mathcal{P}$ is empty if and only if $\langle p_0, \bot \rangle \notin \text{pre}^*(\boldsymbol{R}_\mathsf{S}\Gamma^*)$.

Proof. ($\Rightarrow$) Suppose $\mathcal{P}$ has an accepting run $\rho = c_0 c_1 \cdots$ starting from $\langle p_0, \bot \rangle$ on some infinite word. Let $\{k_1, \ldots, k_n\} \subseteq [k]$ be the set of indices such that for every $i \in [k]$, $i \in \{k_1, \ldots, k_n\}$ if and only if $\mathsf{Inf}(\rho) \cap F_i \neq \emptyset$. Therefore, $\mathsf{Inf}(\rho) \cap F_i = \emptyset$ for every $i \in [k] \setminus \{k_1, \ldots, k_n\}$ and $\mathsf{Inf}(\rho) \cap E_i \neq \emptyset$ for every $i \in \{k_1, \ldots, k_n\}$. Let S be the set of control states $P \setminus (\bigcup_{i \in [k] \setminus \{k_1, \ldots, k_n\}} F_i)$. Since the number of control states is finite, there exists a configuration c_m in ρ such that for every $j \geqslant m$, $c_j \in S \times \Gamma^*$. Let $\rho_{\geqslant m}$ be the suffix $c_m c_{m+1} \cdots$ of ρ. We can construct a subsequence $c_{n_1} c_{n_2} \cdots$ of $\rho_{\geqslant m}$ such that

$$|c_{n_1}| = \min\{|c_j| \mid j \geqslant m\},$$
$$|c_{n_i}| = \min\{|c_j| \mid j > n_{i-1}\}, \ \forall i \geqslant 2.$$

Since the sets S and Γ are finite (i.e., the number of heads is finite), there must exist a pair $\langle p, \gamma \rangle \in S \times \Gamma$ that appears in $c_{n_1} c_{n_2} \cdots$ infinitely often. Therefore, we can construct a subsequence $c_{m_1} c_{m_2} \cdots$ of $c_{n_1} c_{n_2} \cdots$ such that for every $i \geqslant 1$, the head of c_{m_i} is $\langle p, \gamma \rangle$.

By the above construction, there exist two positions $m_{e'} > m_e \geqslant m_1$ such that there is a subsequence $\langle p^{k_1}, u^{k_1} \rangle \cdots \langle p^{k_n}, u^{k_n} \rangle$ of $c_{m_e} c_{m_e+1} \cdots c_{m_{e'}}$ with $p^{k_1} \in E_{k_1}, \ldots, p^{k_n} \in E_{k_n}$. Moreover, for every $i \in [k]$, $\langle p^{k_i}, u^{k_i} \rangle$ is the first configuration from $E_{k_i} \times \Gamma^*$ that occurs in the path $\langle p^{k_{i-1}}, u^{k_{i-1}} \rangle \cdots \langle p^{k_i}, u^{k_i} \rangle$. W.l.o.g., we assume that $c_{m_e} = \langle p, \gamma u_1 \rangle \neq \langle p^{k_1}, u^{k_1} \rangle$ and $c_{m_{e'}} = \langle p, \gamma u_2 u_1 \rangle \neq \langle p^{k_n}, u^{k_n} \rangle$. Therefore, we obtain that $\mathcal{P}_S$ has

$$\langle (p, 0), \gamma u_1 \rangle \overset{w_1}{\Longrightarrow}^+ \langle (p^{k_n}, |\mathsf{In}(S)|), u^{k_n} \rangle \overset{w_2}{\Longrightarrow}^+ \langle (p, 0), \gamma u_2 u_1' \rangle.$$

Since once the configuration c_{m_e} is reached, the rest of the run ρ from the position m_e will never change the stack content in c_{m_e} except for the topmost of the stack, we obtain that $u_1 = u_1'$ and $\mathcal{P}_S$ has

$$\langle (p, 0), \gamma \rangle \overset{w_1}{\Longrightarrow}^+ \langle (p^{k_n}, |\mathsf{In}(S)|), u' \rangle \overset{w_2}{\Longrightarrow}^+ \langle (p, 0), \gamma u_2 \rangle.$$

Therefore, $\langle p, \gamma \rangle \in \boldsymbol{R}_{\mathsf{S}}$. It immediately follows that $\langle p_0, \bot \rangle \in \mathrm{pre}^*(\boldsymbol{R}_{\mathsf{S}} \Gamma^*)$.

($\Leftarrow$) Suppose that $\langle p_0, \bot \rangle \in \mathrm{pre}^*(\boldsymbol{R}_{\mathsf{S}} \Gamma^*)$, let $\langle p, \gamma \rangle \in \boldsymbol{R}_{\mathsf{S}}$ be the head such that $\langle p_0, \bot \rangle \overset{w_1}{\Longrightarrow}{}^* \langle p, \gamma u \rangle$ and $\mathcal{P}_S$ has $\langle (p, 0), \gamma \rangle \overset{w_2}{\Longrightarrow}{}^+ \langle (p', |\mathsf{In}(S)|), u' \rangle \overset{w_3}{\Longrightarrow}{}^+ \langle (p, 0), \gamma u_2 \rangle$ for some set $S \subseteq P$. The following run is an accepting run ρ:

$$\langle p_0, \bot \rangle \overset{w_1}{\Longrightarrow}{}^* \langle p, \gamma u \rangle \overset{w_2}{\Longrightarrow}{}^+ \langle p', u' u \rangle \overset{w_3}{\Longrightarrow}{}^+ \langle p, \gamma u_2 u \rangle \overset{w_2}{\Longrightarrow}{}^+ \langle p', u' u u_2 u \rangle \overset{w_3}{\Longrightarrow}{}^+ \langle p, \gamma u_2 u_2 u \rangle \cdots .$$

Notice that for every $i \in [k]$, if $\mathsf{Inf}(\rho) \cap F_i \neq \emptyset$, then $E_i \in \mathsf{In}(S)$ which implies that $E_i \cap \mathsf{Inf}(\rho) \neq \emptyset$.

For each set $S \subseteq P$, the set R^S can be computed in $\boldsymbol{O}(|S|^2 \cdot |\Delta| \cdot |\mathsf{In}(S)|^3)$ time and $\boldsymbol{O}(|S| \cdot |\Delta| \cdot |\mathsf{In}(S)|^2)$ space by applying Lemma 1. The number of sets S that should be considered is bounded by $2^{\min(k, |P|)}$. Therefore, $\boldsymbol{R}_{\mathsf{S}}$ can be computed in $\boldsymbol{O}(k^3 |P|^2 \cdot |\Delta| \cdot 2^{\min(|P|, k)})$ time and $\boldsymbol{O}(k^2 |P| \cdot |\Delta|)$ space. Applying Theorem 1, we obtain Theorem 5.

Theorem 5. The emptiness problem of SPDAs $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \bot, \mathcal{F})$ can be decided in $\boldsymbol{O}(k^3 |P|^2 \cdot |\Delta| \cdot 2^{\min(|P|, k)})$ time and $\boldsymbol{O}(k^2 |P| \cdot |\Delta|)$ space.

3.4 Efficient solution to the emptiness problem of MPDAs

We fix the MPDA $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \bot, \mathcal{F})$ with $\mathcal{F} = \{F_1, \ldots, F_k\}$. Our approach is based on the observation that the MPDA $\mathcal{P}$ has an accepting run ρ on some infinite word if and only if there is a position $m \geqslant 0$ and an indicator $i \in [k]$ such that for every $n \geqslant m$, the set of control states in the configurations $\rho(n)$ is the set F_i. This observation allows us to reduce the emptiness problem of the MPDA $\mathcal{P}$ to compute heads $\langle p, \gamma \rangle \in F_i \times \Gamma$ of the form $\langle p, \gamma \rangle \overset{w}{\Longrightarrow}{}^+ \langle p, \gamma u \rangle$ such that the set of control states appearing in $\langle p, \gamma \rangle \overset{w}{\Longrightarrow}{}^+ \langle p, \gamma u \rangle$ is F_i for some indicator $i \in [k]$. The later problem is solved by applying Lemma 1 on a BPDA $\mathcal{P}_i$ which counts the number of visited control states in F_i.

Given an index $i \in [k]$, let $\mathcal{P}_i = (\Sigma, F_i \times \{0, \ldots, |F_i|\}, \Gamma, \Delta_i, (p_0, 0), \bot, F_i \times \{|F_i|\})$ be the BPDA, where Δ_i is defined as follows: for every $\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', u \rangle \in \Delta$ with $p, p' \in F_i$, every $j \in \{0, \ldots, |F_i| - 1\}$,

- $\langle (p, j), \gamma \rangle \overset{a}{\hookrightarrow} \langle (p', j + 1), u \rangle \in \Delta_i$, if $p' = p_i^{j+1}$;
- $\langle (p, j), \gamma \rangle \overset{a}{\hookrightarrow} \langle (p', j), u \rangle \in \Delta_i$;
- $\langle (p, |F_i|), \gamma \rangle \overset{\epsilon}{\hookrightarrow} \langle (p', 0), u \rangle \in \Delta_i$.

Let $\boldsymbol{R}_{\mathsf{B}}^i$ be the set of repeating heads in the BPDA $\mathcal{P}_i$, $\boldsymbol{R}_{\mathsf{M}}$ be the set $\{\langle p, \gamma \rangle \in P \times \Gamma \mid \langle (p, 0), \gamma \rangle \in \boldsymbol{R}_{\mathsf{B}}^i, i \in [k]\}$.

Proposition 4. The MPDA $\mathcal{P}$ is empty if and only if $\langle p_0, \bot \rangle \notin \mathrm{pre}^*(\boldsymbol{R}_{\mathsf{M}} \Gamma^*)$.

Proof. ($\Rightarrow$) Suppose that $\mathcal{P}$ has an accepting run $\rho = c_0 c_1 \cdots$ on the infinite word $w = a_0 a_1 \cdots$, where $c_0 = \langle p_0, \bot \rangle$. Let $F_m \in \mathcal{F}$ be the set of control states such that $\mathsf{Inf}(\rho) = F_m$. Then, there exists a position $m' \geqslant 0$ such that the control states appearing in the sequence $\rho_{\geqslant m'} = c_{m'} c_{m'+1} \cdots$ are in F_m. We can construct a subsequence $c_{n_1} c_{n_2} \cdots$ of $\rho_{\geqslant m'}$ such that

$$|c_{n_1}| = \min\{|c_j| \mid j \geqslant m'\},$$
$$|c_{n_i}| = \min\{|c_j| \mid j > n_{i-1}\}, \ \forall i \geqslant 2.$$

Since the sets F_m and Γ are finite (i.e., the number of heads is finite), there must exist a pair $\langle p, \gamma \rangle \in F_m \times \Gamma$ that appears in $c_{n_1} c_{n_2} \cdots$ infinitely often. Therefore, we can construct a subsequence $c_{j_1} c_{j_2} \cdots$ of $c_{n_1} c_{n_2} \cdots$ such that for every $i \geqslant 1$, the head of c_{j_i} is $\langle p, \gamma \rangle$.

By the above construction, there exist two positions $j_{e'} > j_e \geqslant j_1$ such that $\mathcal{P}_m$ has

$$c_{j_e} = \langle (p, 0), \gamma u_1 \rangle \overset{w_1}{\Longrightarrow}{}^+ \langle (p', |F_m|), v u_1 \rangle \overset{w_2}{\Longrightarrow}{}^+ \langle (p, 0), \gamma u_2 u_1' \rangle = c_{j_{e'}}.$$

Considering that once the configuration c_{j_e} is reached, the rest of the run ρ from the position j_e will never change the stack content in c_{j_e} except for the topmost of the stack, we obtain that $u_1 = u_1'$ and $\mathcal{P}_m$ has

$$\langle (p, 0), \gamma \rangle \overset{w_1}{\Longrightarrow}{}^+ \langle (p', |F_m|), v \rangle \overset{w_2}{\Longrightarrow}{}^+ \langle (p, 0), \gamma u_2 \rangle.$$

Therefore, $\langle p, \gamma \rangle \in \boldsymbol{R}_{\mathsf{M}}$. The result immediately follows.

($\Leftarrow$) Suppose that $\langle p_0, \bot \rangle \in \mathrm{pre}^*(\boldsymbol{R}_{\mathsf{M}}\Gamma^*)$. Let $\langle p, \gamma \rangle \in \boldsymbol{R}_{\mathsf{M}}$ be the head such that

$$\langle p_0, \bot \rangle \overset{w_0}{\Longrightarrow}^* \langle p, \gamma u \rangle \text{ and } \mathcal{P}_i \text{ has } \langle (p, 0), \gamma \rangle \overset{w_1}{\Longrightarrow}^+ \langle (p', |F_i|), v \rangle \overset{w_2}{\Longrightarrow}^+ \langle (p, 0), \gamma u' \rangle$$

for some $i \in [k]$. Therefore, we obtain the following run:

$$\langle p_0, \bot \rangle \overset{w_0}{\Longrightarrow}^* \langle p, \gamma u \rangle \overset{w_1}{\Longrightarrow}^+ \langle p', vu \rangle \overset{w_2}{\Longrightarrow}^+ \langle p, \gamma u' u \rangle \overset{w_1}{\Longrightarrow}^+ \langle p', vu'u \rangle \overset{w_2}{\Longrightarrow}^+ \langle p, \gamma u'u'u \rangle \cdots ,$$

where only the set F_i of control states are visited infinitely often after $\langle p, \gamma u \rangle$.

Applying Lemma 1, the set $\boldsymbol{R}_{\mathsf{B}}^i$ can be computed in $\boldsymbol{O}(|F_i|^5 \cdot |\Delta|)$ time and $\boldsymbol{O}(|\Delta| \cdot |F_i|^3)$ space. This implies that $\boldsymbol{R}_{\mathsf{M}}$ can be computed in $\boldsymbol{O}(|\Delta| \cdot \sum_{i \in [k]} |F_i|^5)$ time and $\boldsymbol{O}(|\Delta| \cdot \max\{|F_i|^3 \mid i \in [k]\})$ space. Notices that once $\boldsymbol{R}_{\mathsf{B}}^i$ is computed, the space used for computing $\boldsymbol{R}_{\mathsf{B}}^i$ can be deallocated. Applying Theorem 1, we obtain Theorem 6.

Theorem 6. The emptiness problem of MPDAs $\mathcal{P} = (\Sigma, P, \Gamma, \Delta, p_0, \bot, \mathcal{F})$ can be decided in $\boldsymbol{O}(|\Delta| \cdot \max(|P|^2, \sum_{i \in [k]} |F_i|^5))$ time and $\boldsymbol{O}(|\Delta| \cdot \max\{|P|, |F_i|^3 \mid i \in [k]\})$ space.

The time complexity $\boldsymbol{O}(|\Delta| \cdot \max(|P|^2, \sum_{i \in [k]} |F_i|^5))$ is obtained from the fact that $\boldsymbol{R}_{\mathsf{M}}$ can be computed in $\boldsymbol{O}(|\Delta| \cdot \sum_{i \in [k]} |F_i|^5)$ time and $\mathrm{pre}^*(\boldsymbol{R}_{\mathsf{M}}\Gamma^*)$ is computed in $\boldsymbol{O}(|\Delta| \cdot |P|^2)$ time by Theorem 1. Similarly, the space complexity $\boldsymbol{O}(|\Delta| \cdot \max\{|P|, |F_i|^3 \mid i \in [k]\})$ comes from the fact that $\boldsymbol{R}_{\mathsf{M}}$ can be computed in $\boldsymbol{O}(|\Delta| \cdot \max\{|P|, |F_i|^3 \mid i \in [k]\})$ space and $\mathrm{pre}^*(\boldsymbol{R}_{\mathsf{M}}\Gamma^*)$ is computed in $\boldsymbol{O}(|\Delta| \cdot |P|)$ space by Theorem 1.

The emptiness algorithm presented in this section has a tighter upper bound than the one that goes through BPDAs. In particular, in the worst-case when the maximum set in $\mathcal{F}$ is P and $|\mathcal{F}| = 2^{|P|}$, this efficient approach works in $\boldsymbol{O}(|P|^5 \cdot |\Delta| \cdot 2^{|P|})$ time and $\boldsymbol{O}(|P|^3 \cdot |\Delta|)$ space, while the latter works in $\boldsymbol{O}(|P|^5 \cdot |\Delta| \cdot 2^{3|P|})$ time and $\boldsymbol{O}(|P|^3 \cdot |\Delta| \cdot 2^{2|P|})$ space.

3.5 Lower bounds

It was already known that the context-free-language reachability problem is PTIME-complete [34]. This implies that the control state reachability problem of PDSs is PTIME-complete. The control state reachability problem can be reduced to the emptiness problem of ω-PDAs in polynomial time by adding a self-loop on the target control state and adjusting the acceptance condition. Therefore, the emptiness problem of ω-PDAs is PTIME-hard. From Theorems 2, 4–6, we obtain that the emptiness problems of BPDAs, PPDAs, RPDAs and MPDAs are PTIME-complete, while the problem of SPDAs is PTIME-hard. It remains open whether there exists an optimal algorithm for SPDAs. We leave this to future work.

4 Model checking linear temporal logic with fairness

In this section, we demonstrate one of the benefits of our efficient algorithms by applying it to the model checking problem for linear temporal logic on PDAs while considering fairness. We found that it is better to verify LTL formulae on PDSs by reducing to the emptiness problem of SPDAs rather than that of BPDAs in term of space complexity.

4.1 Linear temporal logic and Büchi automata

Let us fix the PDA $\mathcal{P} = (P, \Gamma, \Delta, p_0, \bot)$ in this section. Let AP be a finite set of atomic propositions.

Definition 3. Formulae of the Linear Temporal Logic (LTL for short) are defined by the following rules:

$$\phi \ ::= \ \sigma \ | \ \neg\phi \ | \ \phi \wedge \phi \ | \ X \ | \ \phi U \phi,$$

where $\sigma \in$ AP.

Given an infinite word $w = w_0 w_1 \cdots \in (2^{\mathsf{AP}})^\omega$, let $w^i = w_i w_{i+1} \cdots$ be the suffix of w. The satisfaction relation $w \models \phi$ between an infinite word w and an LTL formula ϕ is inductively defined as follows:

$$w \models \sigma \iff \sigma \in w_0;$$
$$w \models \neg\phi \iff w \not\models \phi;$$
$$w \models \phi_1 \wedge \phi_2 \iff w \models \phi_1 \text{ and } w \models \phi_2;$$
$$w \models X\phi \iff w^1 \models \phi;$$
$$w \models \phi_1 U \phi_2 \iff \exists i \geqslant 0 \text{ such that } w^i \models \phi_2 \text{ and } \forall j : 0 \leqslant j < i, \ w^j \models \phi_1.$$

We use abbreviations $F\phi \equiv \mathrm{true} U \phi$, $G\phi \equiv \neg F\neg\phi$ and $\phi_1 \to \phi_2 \equiv \neg\phi_1 \vee \phi_2$. Let $l : \mathsf{AP} \to 2^P$ be a labeling function that assigns to each atomic proposition a set of control states. Given a run $\rho = c_0 c_1 \cdots$ of $\mathcal{P}$, let $l(\rho)$ be the infinite word w such that for every $i \geqslant 0$, $w_i = \{\sigma \in \mathsf{AP} \mid c_i \in l(\sigma) \times \Gamma^*\}$. A run $\rho = c_0 c_1 \cdots$ of $\mathcal{P}$ satisfies an LTL formula ϕ if and only if $l(\rho) \models \phi$. A configuration c satisfies an LTL formula ϕ if and only if for each run ρ of $\mathcal{P}$ starting from c, ρ satisfies ϕ. Let $\|\phi\|^{\mathcal{P}}$ be the set of configurations that satisfy the formula ϕ.

The local model checking problem for LTL is to decide whether $c \in \|\phi\|^{\mathcal{P}}$ for a given LTL formula ϕ and a configuration $c \in \mathbb{C}$, while the global model checking problem is to compute $\|\phi\|^{\mathcal{P}}$. In this work, we solve the global model checking problem using an automata-theoretic approach that reduces the problem to the emptiness problem of ω-PDAs.

Definition 4. A Büchi-automaton (BA) $\boldsymbol{M}$ is a tuple $(G, \Sigma, \delta, g^0, \mathcal{F})$ where G is a finite set of states, Σ is the input alphabet, $\delta : G \times \Sigma \to 2^G$ is a transition function, $g^0 \in G$ is the initial state and $\mathcal{F} \subseteq G$ is the Büchi acceptance.

A run π of $\boldsymbol{M}$ over an infinite word $\alpha_0 \alpha_1 \cdots \in \Sigma^\omega$ is an infinite sequence of states $\pi = g_0 g_1 \cdots$ such that $g_0 = g^0$, and for every $i \geqslant 0$, $g_{i+1} \in \delta(g_i, \alpha_i)$. Let $\mathsf{Inf}(\pi)$ be the set of states visited infinitely often in π. A run π is accepting if and only if $\mathsf{Inf}(\pi) \cap \mathcal{F} \neq \emptyset$.

Theorem 7 ([10]).   For every LTL formula ϕ, we can construct a BA $\boldsymbol{M} = (G, 2^{\mathsf{AP}}, \delta, g^0, F)$ with $2^{O(|\phi|)}$ states/transitions such that $\boldsymbol{M}$ recognizes all of the infinite words satisfying ϕ, where $|\phi|$ denotes the number of subformulae of ϕ.

Theorem 8 ([24]).   Given a PDS $\mathcal{P} = (P, \Gamma, \Delta, p_0, \bot)$ and an LTL formula ϕ, $\|\phi\|^{\mathcal{P}}$ can be computed in $O(|P|^2 \cdot |\Delta| \cdot 2^{O(|\phi|)})$ time and $O(|P| \cdot |\Delta| \cdot 2^{O(|\phi|)})$ space.

4.2   Fairness

Fairness is the key to concurrent systems which rules out infinite behaviors that are considered unrealistic, and are often necessary to establish liveness properties frequently encountered in concurrent systems. Fairness assumption can be expressed in an LTL formula. In this work, we consider the following fairness assumptions [35]:

$$\text{Unconditional fairness} : \Phi \equiv \bigwedge_{1 \leqslant i \leqslant k} GF\phi_i,$$

$$\text{Weak fairness} : \Phi \equiv \bigwedge_{1 \leqslant i \leqslant k} (FG\phi_i \to GF\psi_i),$$

$$\text{Strong fairness} : \Phi \equiv \bigwedge_{1 \leqslant i \leqslant k} (GF\phi_i \to GF\psi_i),$$

where ϕ_i and ψ_i are LTL formulae.

For the sake of simplicity, we assume that the formulae ϕ_i and ψ_i in fairness assumptions are Boolean combinations of atomic propositions. Given a control state $p \in P$ and the labeling function l, $p \models \psi_i$ denotes that ψ_1 holds when replacing each atomic proposition σ in ψ_1 by true (resp. false) if $p \in l(\sigma)$ (resp. $p \notin l(\sigma)$). Our approach can be generalized to cope with general fairness assumptions using regular valuations [36].

For a PDA $\mathcal{P} = (P, \Gamma, \Delta, p_0, \bot)$ and an LTL formula ϕ with the fairness assumption Φ, then the global model checking problem is to compute $\|\Phi \to \phi\|^{\mathcal{P}}$. By applying Theorem 8, $\|\Phi \to \phi\|^{\mathcal{P}}$ can be computed in $\boldsymbol{O}(|P|^2 \cdot |\Delta| \cdot 2^{\boldsymbol{O}(|\phi|+k)})$ time and $\boldsymbol{O}(|P| \cdot |\Delta| \cdot 2^{\boldsymbol{O}(|\phi|+k)})$ space. In the next section, we will present an alternative approach by reducing the model checking problem to the emptiness problem of SPDAs.

It is well-known that weak fairness $\bigwedge_{1 \leqslant i \leqslant k}(FG\phi_i \to GF\psi_i)$ can be written into an unconditional fairness $\bigwedge_{1 \leqslant i \leqslant k} GF(\neg\phi_i \vee \psi_i)$ and unconditional fairness $\bigwedge_{1 \leqslant i \leqslant k} GF\phi_i$ can be seen as a special form of strong fairness $\bigwedge_{1 \leqslant i \leqslant k}(GF \text{ true} \to GF\phi_i)$. Therefore, it is adequate to consider strong fairness.

4.3 A space-efficient model checking approach for LTL with fairness

Inspired by the model checking approach for LTL on PDSs which is reduced to the emptiness problem of BPDAs, and fairness Φ can be transformed into some Streett acceptance, we can reduce the LTL model checking problem on PDSs with fairness to the emptiness problem of SPDAs.

Fix a PDS $\mathcal{P} = (P, \Gamma, \Delta, p_0, \bot)$ and an LTL formula ϕ with the strong fairness assumption $\Phi \equiv \bigwedge_{1 \leqslant i \leqslant k}(GF\phi_i \to GF\psi_i)$. Let $\boldsymbol{M}_{\neg\phi} = (G, 2^{\mathsf{AP}}, \delta, g^0, F)$ be the NBA that recognizes all of the infinite words satisfying $\neg\phi$. We define a SPDA $\mathcal{P}_{\neg\phi} = (\{a\}, P \times G, \Gamma, \Delta', [p_0, g^0], \bot, \mathcal{F})$, where

- Δ' is defined as follows: for every $\langle p, \gamma \rangle \hookrightarrow \langle p', u \rangle \in \Delta$ and every $g' \in \delta(g, A)$ with $A = \{\sigma \in \mathsf{AP} \mid p \in l(\sigma)\}$, $\langle [p, g], \gamma \rangle \overset{a}{\hookrightarrow} \langle [p', g'], u \rangle \in \Delta'$,
- $\mathcal{F} = \{(E_1, F_1), \ldots, (E_{k+1}, F_{k+1})\}$, where for every $i : 1 \leqslant i \leqslant k$, $E_i = \{p \in P \mid p \models \psi_i\} \times G$ and $F_i = \{p \in P \mid p \models \phi_i\} \times G$, $F_{k+1} = P \times G$ and $E_{k+1} = F \times G$.

Lemma 3. For every configuration $\langle p, u \rangle \in P \times \Gamma^*$, $\langle p, u \rangle \notin \|\Phi \to \phi\|^{\mathcal{P}}$ if and only if $\mathcal{P}_{\neg\phi}$ has an accepting run starting from $\langle [p, g^0], u \rangle$.

Proof. Suppose that $\langle p, u \rangle \notin \|\Phi \to \phi\|^{\mathcal{P}}$, then $\mathcal{P}$ has a run ρ starting from $\langle p, u \rangle$ such that $\rho \models \Phi$, but $\rho \not\models \phi$. The later implies that $\rho \models \neg\phi$, i.e., $\boldsymbol{M}_{\neg\phi}$ has an accepting run π on the infinite word $l(\rho)$. Let $\rho_\pi \in (P \times G \times \Gamma^*)^\omega$ be a run such that the projection of ρ_π on $\mathcal{P}$ is the run ρ, the projection of ρ_π on G is π. It is easy to see that ρ_π is an accepting run of $\mathcal{P}_{\neg\phi}$.

Suppose $\mathcal{P}_{\neg\phi}$ has an accepting run ρ starting from $\langle [p, g^0], u \rangle$, let $\rho_{\mathcal{P}}$ be the projection of ρ on $\mathcal{P}$ and ρ_G be the projection of ρ on G. For every $i : 1 \leqslant i \leqslant k$, since either $F_i \cap \mathsf{Inf}(\rho) = \emptyset$ or $E_i \cap \mathsf{Inf}(\rho) \neq \emptyset$ holds, then $\rho_{\mathcal{P}} \models \Phi$. On the other hand, it is easy to verify that ρ_G is an accepting run of $\mathcal{P}_{\neg\phi}$ over the infinite word $l(\rho_{\mathcal{P}})$ and $\rho_{\mathcal{P}}$ is a run of $\mathcal{P}$. The result immediately follows.

Applying Theorem 5, we obtain Theorem 9.

Theorem 9. Given an LTL formula ϕ and a PDS $\mathcal{P} = (P, \Gamma, \Delta, p_0, \bot)$ with the fairness Φ, $\|\Phi \to \phi\|^{\mathcal{P}}$ can be computed in
- $\boldsymbol{O}(k^3|P|^2 \cdot |\Delta| \cdot 2^{\boldsymbol{O}(|\phi|)+\min(|P|,k)})$ time and $\boldsymbol{O}(k^2|P| \cdot |\Delta| \cdot 2^{\boldsymbol{O}(|\phi|)})$ space if Φ is a strong fairness,
- $\boldsymbol{O}(k^3|P|^2 \cdot |\Delta| \cdot 2^{\boldsymbol{O}(|\phi|)})$ time and $\boldsymbol{O}(k^2|P| \cdot |\Delta| \cdot 2^{\boldsymbol{O}(|\phi|)})$ space if Φ is a weak or unconditional fairness.

The complexity for weak/unconditional fairness is obtained by a careful analysis of the algorithm for emptiness checking of SPDAs and the fact that the Streett acceptance is $\mathcal{F} = \{(E_1, P \times G), \ldots, (E_{k+1}, P \times G)\}$, where for every $i : 1 \leqslant i \leqslant k$, $E_i = \{p \in P \mid p \models \psi_i\} \times G$ and $E_{k+1} = F \times G$ (which is a generalized Büchi acceptance).

Compared to the approach via BPDAs, our approach has the following advantages. Our approach reduces the space complexity from an exponential time of k to a polynomial time of k, as well as the time complexity for weak and unconditional fairness. For strong fairness, our approach has a tight time complexity when k is greater than $|P|$.

5 Conclusion and future work

In this paper, we proposed direct and efficient algorithms for the emptiness problems of PPDAs, RPDAs, SPDAs and MPDAs, which are crucial for model checking recursive programs with fairness. In these recursive programs with fairness, one thread is recursive, the other threads are finite and all the threads are executed in some "fair" manner.

Future work will include the implementation of proposed algorithms, investigation of ω-PDAs in which the sets in acceptance conditions are sets of configurations rather than control states, and investigation of the emptiness problem of alternating ω-PDAs (ω-APDAs). Although, the emptiness problem is undecidable [37] in general, the problem for alphabet-free ω-APDAs with Büchi or Parity acceptance was already investigated in many studies [27, 29–31, 38, 39]. There is no direct approach to the emptiness problem for alphabet-free ω-APDAs with other acceptances. Along these lines, it will also be interesting to investigate efficient algorithms for the emptiness problem of alphabet-free ω-APDAs with related acceptances, which are related to the pushdown model checking problem against branching-time temporal logic with fairness.

Conflict of interest The authors declare that they have no conflict of interest.

References

1　Büchi J R. On a decision method in restricted second order arithmetic. In: The Collected Works of J. Richard Büchi. New York: Springer-Verlag, 1990. 425–435

2　Muller D E. Infinite sequences and finite machines. In: Proceedings of the 4th Annual Symposium on Switching Circuit Theory and Logical Design. New York: IEEE, 1963. 3–16

3　McNaughton R. Testing and generating infinite sequences by a finite automaton. Inf Contr, 1966, 9: 521–530

4　Rabin M O. Decidability of second-order theories and automata on infinite trees. Trans Amer Math Soc, 1969, 141: 1–35

5　Streett R S. Propositional dynamic logic of looping and converse is elementary decidable. Inf Contr, 1982, 54: 121–141

6　Mostowski A W. Regular expressions for infinite trees and a standard form of automata. In: Proceedings of the 5th Symposium on Computation Theory. Berlin: Springer-Verlag, 1984. 157–168

7　Emerson E A, Lei C. Modalities for model checking: branching time strikes back. In: Proceedings of the 12th ACM Symposium on Principles of Programming Languages. New York: ACM, 1985. 84–96

8　Thomas W. Automata on infinite objects. In: Handbook of Theoretical Computer Science, Volume B. Cambridge: MIT Press, 1990. 133–192

9　Vardi M Y. Verification of concurrent programs: the automata-theoretic framework. Ann Pure Appl Logic, 1991, 51: 79–98

10　Vardi M Y, Wolper P. Automata-theoretic techniques for modal logics of programs. J Comput Syst Sci, 1986, 32: 183–221

11　Pnueli A. The temporal logic of programs. In: Proceedings of the 18th Annual Symposium on Foundations of Computer Science. New York: IEEE, 1977. 46–57

12　Löding C. Methods for the transformation of ω-automata: complexity and connection to second order logic. Dissertation for Ph.D. Degree. Kiel: Christian-Albrechts-University of Kiel, 1998

13　Löding C. Optimal bounds for transformations of omega-automata. In: Proceedings of the 19th Conference on Foundations of Software Technology and Theoretical Computer Science. Berlin: Springer-Verlag, 1999. 97–109

14　Piterman N. From nondeterministic Büchi and Streett automata to deterministic parity automata. Log Meth Comput Sci, 2007, 3: 1–21

15　Bloem R, Gabow H N, Somenzi F. An algorithm for strongly connected component analysis in $n \log n$ symbolic steps. Form Meth Syst Des, 2006, 28: 37–56

16　Henzinger M R, Telle J A. Faster algorithms for the nonemptiness of Streett automata and for communication protocol pruning. In: Proceedings of the 5th Scandinavian Workshop on Algorithm Theory. Berlin: Springer-Verlag, 1996. 16–27

17　King V, Kupferman O, Vardi M Y. On the complexity of parity word automata. In: Proceedings of the 17th International Conference on Foundations of Software Science and Computation Structures. Berlin: Springer-Verlag, 2001. 276–286

18　Cohen R S, Gold A Y. Theory of omega-languages I: characterizations of omega-context-free languages. J Comput Syst Sci, 1977, 15: 169–184

19　Cohen R S, Gold A Y. Theory of omega-languages II: a study of various models of omega-type generation and recognition. J Comput Syst Sci, 1977, 15: 185–208

20　Cohen R S, Gold A Y. Omega-computations on deterministic pushdown machines. J Comput Syst Sci, 1978, 16: 275–300

21　Linna M. On omega-sets associated with context-free languages. Inf Contr, 1976, 31: 272–293

22　Linna M. A decidability result for deterministic ω-context-free languages. Theor Comput Sci, 1977, 4: 83–98

23 Bouajjani A, Esparza J, Maler O. Reachability analysis of pushdown automata: application to model checking. In: Proceedings of the 8th International Conference on Concurrency Theory. Berlin: Springer-Verlag, 1997. 135–150

24 Esparza J, Hansel D, Rossmanith P, et al. Efficient algorithm for model checking pushdown systems. In: Proceedings of the 12th International Conference Computer Aided Verification. Berlin: Springer-Verlag, 2000. 232–247

25 Esparza J, Schwoon S. A BDD-based model checker for recursive programs. In: Proceedings of the 13th International Conference Computer Aided Verification. Berlin: Springer-Verlag, 2001. 324–336

26 Finkel A, Willems B, Wolper P. A direct symbolic approach to model checking pushdown systems. Electr Notes Theor Comput Sci, 1997, 9: 27–37

27 Walukiewicz I. Pushdown processes: games and model checking. Inf Comput, 1996, 164: 62–74

28 Hague M, Ong C L. Winning regions of pushdown parity games: a saturation method. In: Proceedings of the 20th International Conference on Concurrency Theory. Berlin: Springer-Verlag, 2009. 384–398

29 Hague M, Ong C L. A saturation method for the modal μ-calculus over pushdown systems. Inf Comput, 2011, 209: 799–821

30 Kupferman O, Piterman N, Vardi M Y. Pushdown specifications. In: Proceedings of the 9th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning. Berlin: Springer-Verlag, 2002. 262–277

31 Song F, Touili T. Efficient CTL model-checking for pushdown systems. In: Proceedings of the 8th International Conference on Concurrency Theory. Berlin: Springer-Verlag, 2011. 434–449

32 Song F, Touili T. Efficient CTL model-checking for pushdown systems. Theor Comput Sci, 2014, 549: 127–145

33 Song F, Touili T. PoMuC: a CTL model-checker for sequential programs. In: Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering. New York: ACM, 2012. 346–349

34 Reps T W. Program analysis via graph reachability. Inf Softw Tech, 1998, 40: 701–726

35 Baier C, Katoen J. Principles of Model Checking. London: MIT Press, 2008

36 Esparza J, Kucera A, Schwoon S. Model checking LTL with regular valuations for pushdown systems. Inf Comput, 2003, 186: 355–376

37 Bozzelli L. Complexity results on branching-time pushdown model checking. Theor Comput Sci, 2007, 379: 286–297

38 Cachat T. Symbolic strategy synthesis for games on pushdown graphs. In: Proceedings of the 29th International Colloquium on Automata, Languages and Programming. Berlin: Springer-Verlag, 2002. 704–715

39 Serre O. Note on winning positions on pushdown games with omega-regular conditions. Inf Process Lett, 2003, 85: 285–291