MODEL-CHECKING OF ORDERED MULTI-PUSHDOWN AUTOMATA [*]

MOHAMED FAOUZI ATIG

Uppsala University, Sweden
e-mail address: mohamed_faouzi.atig@it.uu.se

ABSTRACT. We address the verification problem of ordered multi-pushdown automata: A multi-stack extension of pushdown automata that comes with a constraint on stack transitions such that a pop can only be performed on the first non-empty stack. First, we show that the emptiness problem for ordered multi-pushdown automata is in 2ETIME. Then, we prove that, for an ordered multi-pushdown automata, the set of all predecessors of a regular set of configurations is an effectively constructible regular set. We exploit this result to solve the global model-checking which consists in computing the set of all configurations of an ordered multi-pushdown automaton that satisfy a given w-regular property (expressible in linear-time temporal logics or the linear-time μ-calculus). As an immediate consequence, we obtain an 2ETIME upper bound for the model-checking problem of w-regular properties for ordered multi-pushdown automata (matching its lower-bound).

INTRODUCTION

Automated verification of multi-threaded programs is an important and a highly challenging problem. In fact, even when such programs manipulate data ranging over finite domains, their control structure can be complex due to the handling of (recursive) procedure calls in the presence of concurrency and synchronization between threads.

In the last few years, a lot of effort has been devoted to the verification problem for models of concurrent programs (see, e.g., [BMOT05, TMP07, Kah09, ABT08, TMP08, AT09, HLMS10, LR08, GMM10, EQR11, BEP11, LN11]) where each thread corresponds to a sequential program with (recursive) procedure calls. In fact, it is well admitted that pushdown automata are adequate models for such kind of threads [EK99, RSJ03], and therefore, it is natural to model recursive concurrent programs as multi-stack automata.

In general, multi-stack automata are Turing powerful and hence come along with undecidability of basic decision problems [Ram00]. A lot of efforts have been nevertheless devoted recently to the development of precise analysis algorithms of specific formal models of some classes of programs [LS98, EP00, BT03, SV06, JM07].

Context-bounding has been proposed in [QR05] as a suitable technique for the analysis of multi-stack automata. The idea is to consider only runs of the automaton that can be

divided into a given number of contexts, where in each context pop and push transitions are exclusive to one stack. The state space which may be explored is still unbounded in presence of recursive procedure calls, but the context-bounded reachability problem is NP-complete even in this case. In fact, context-bounding provides a very useful tradeoff between computational complexity and verification coverage.

In [TMP07], La Torre et al. propose a more general definition of the notion of a context. For that, they define the class of *bounded-phase visibly multi-stack pushdown automata* (BVMPA) where only those runs are taken into consideration that can be split into a given number of phases, where each phase admits pop transitions of one particular stack only. In the above case, the emptiness problem is decidable in double exponential time by reducing it to the emptiness problem for tree automata.

Another way to regain decidability is to impose some order on stack transitions. In [BCCC96], Breveglieri et al. define *ordered multi-pushdown automata* (OMPA), which impose a linear ordering on stacks. Stack transitions are constrained in such a way that a pop transition is reserved to the first non-empty stack. In [ABH08], the emptiness problem for OMPA is shown to be 2ETIME-complete. (Recall that 2ETIME is the class of all decision problems solvable by a deterministic Turing machine in time $2^{2^{dn}}$ for some constant d.) The proof of this result lies in an encoding of OMPA into some class of grammars for which the emptiness problem is decidable. Moreover, the class of ordered multi-pushdown automata with $2k$ stacks is shown to be strictly more expressive than bounded-phase visibly multi-stack pushdown automata with k phases [ABH08].

In this paper, we consider the problem of verifying ordered multi-pushdown automata with respect to a given w-regular property (expressible in the linear-time temporal logics [Pnu77] or the linear-time μ-calculus [Var88]). In particular, we are interested in solving the global model checking for ordered multi-pushdown automata which consists in computing the set of all configurations that satisfy a given w-regular property. The basic ingredient for achieving this goal is to define a procedure for computing the set of backward reachable configurations from a given set of configurations. Therefore, our first task is to find a finite symbolic representation of the possibly infinite state-space of an ordered multi-pushdown automaton. For that, we consider the class of recognizable sets of configurations defined using finite state automata [QR05, ABT08, Set10].

We show that for an ordered multi-pushdown automaton $\mathcal{M}$ the set of all predecessors $Pre^*(C)$ of a recognizable set of configurations C is an effectively constructible recognizable set. For this, we introduce the class of *effective generalized pushdown automata* (EGPA) where transitions on stacks are (1) pop the top symbol of the stack, and (2) push a word in some *effective language* L over the stack alphabet. The language L is said to be *effective* if the problem consisting in checking whether L intersects a given regular language is decidable. Observe that L can be any finite union of languages defined by a class of automata closed under intersection with regular languages and for which the emptiness problem is decidable (e.g., pushdown automata, Petri nets, lossy channel machines, etc). Then, we show that the automata-based saturation procedure for computing the set of predecessors in standard pushdown automata [BEM97] can be extended to prove that for EGPA too the set of all predecessors of a regular set of configurations is a regular set and effectively constructible. As an immediate consequence of this result, we obtain similar decidability results of the decision problems for EGPA like the ones obtained for pushdown automata.

Then, we show that, given an OMPA $\mathcal{M}$ with n stacks, it is possible to construct an EGPA $\mathcal{P}$, whose pushed languages are defined by OMPA with $(n-1)$ stacks, such that

the following invariant is preserved: The state and the stack content of $\mathcal{P}$ are respectively the same as the state and the content of the nth stack of $\mathcal{M}$ when its first $(n-1)$ stacks are empty. Let C be a recognizable set of configurations of $\mathcal{M}$, and $Pre^*(C)$ the set of predecessors of C. Then, we can apply the saturation procedure to $\mathcal{P}$ to show that the set of configurations C_n, consisting of $Pre^*(C)$ restricted to the configurations in which the first $(n-1)$ empty stacks are empty, is recognizable and effectively constructible. To compute the intermediary configurations in $Pre^*(C)$ where the first $(n-1)$ stacks are not empty, we construct an ordered multi-pushdown automaton $\mathcal{M}'$ with $(n-1)$ stacks that: (1) performs the same transitions on its stacks as the ones performed by $\mathcal{M}$ on its first $(n-1)$ stacks, and (2) simulates a push transition of $\mathcal{M}$ over its nth stack by a transition of the finite-state automaton accepting the recognizable set of configurations C_n. Now, we can apply the induction hypothesis to $\mathcal{M}'$ and construct a finite-state automaton accepting the set of all predecessors $Pre^*(C)$.

As an application of this result, we show that the set of configurations of an ordered multi-pushdown automaton satisfying a given w-regular property is recognizable and effectively constructible. Our approach also allows us to obtain an 2ETIME upper bound for the model checking problem of w-regular properties for ordered multi-pushdown automata (matching its lower-bound [ABH08]).

Related works: As mentioned earlier, context-bounding has been introduced by Qadeer and Rehof in [QR05] for detecting safety bugs in shared memory concurrent programs. Several extensions of context-bounding to other classes of programs and efficient procedures for context-bounded analysis have been proposed in [BESS05, BFQ07, LR08, ABQ09, TMP09, LMP09, LMP10]. Other bounding concepts allowing for larger/incomparable coverage of the explored behaviors have been proposed in [TMP07, GMM10, EQR11, BEP11, LN11].

In [Set10], A. Seth shows that the set of predecessors of a recognizable set of configurations of a bounded-phase visibly multi-stack pushdown automaton is recognizable and effectively constructible. In fact, our results generalize the obtained result in [Set10] since any bounded-phase visibly multi-stack pushdown automaton with k phases can be simulated by an ordered multi-pushdown automaton with $2k$ stacks [ABH08].

In this line of work, the focus has been on checking safety properties. In [MP11], P. Madhusudan and G. Parlato propose a unified and generalized technique to show the decidability of the emptiness problem for several restricted classes of concurrent pushdown automata (including ordered multi-pushdown automata). The proof is done by showing that the graphs of each such computations (seen as a multi-nested words) have a bounded tree-width. This result implies that model checking MSO properties (over finite-computations) for these systems is decidable for OMPA. In the conclusion of [MP11], the authors claim that their approach can be used to show the decidability of the model checking of ω-regular properties over infinite computations of OMPA but no proof was provided. Moreover, the authors does not address the global model-checking problem for OMPA neither establish its complexity as we do.

To the best of our knowledge, this is the first work that addresses the global model checking for ordered multi-pushdown automata. In this paper, we extend [Ati10a, Ati10b] by adding details and missing proofs.

1. Preliminaries

In this section, we introduce some basic definitions and notations that will be used in the rest of the paper.

Integers: Let $\mathbb{N}$ be the set of natural numbers. For every $i, j \in \mathbb{N}$ such that $i \leq j$, we use $[i, j]$ (resp. $[i, j[$) to denote the set $\{k \in \mathbb{N} \mid i \leq k \leq j\}$ (resp. $\{k \in \mathbb{N} \mid i \leq k < j\}$).

Words and languages: Let Σ be a finite alphabet. We denote by Σ^* (resp. Σ^+) the set of all words (resp. non empty words) over Σ, and by ϵ the empty word. A language is a (possibly infinite) set of words. We use Σ_ϵ and $Lang(\Sigma)$ to denote respectively the set $\Sigma \cup \{\epsilon\}$ and the set of all languages over Σ. Let u be a word over Σ. The length of u is denoted by $|u|$. For every $j \in [1, |u|]$, we use $u(j)$ to denote the jth letter of u. We denote by u^R the mirror of u.

Transition systems: A transition system (TS for short) is a triplet $\mathcal{T} = (C, \Sigma, \rightarrow)$ where: (1) C is a (possibly infinite) set of configurations, (2) Σ is a finite set of labels (or actions) such that $C \cap \Sigma = \emptyset$, and (3) $\rightarrow \subseteq C \times \Sigma_\epsilon \times C$ is a transition relation. We write $c \xrightarrow{a}_\mathcal{T} c'$ whenever c and c' are two configurations and a is an action such that $(c, a, c') \in \rightarrow$.

Given two configurations $c, c' \in C$, a finite run ρ of $\mathcal{T}$ from c to c' is a finite sequence $c_0 a_1 c_1 \cdots a_n c_n$, for some $n \geq 1$, such that: (1) $c_0 = c$ and $c_n = c'$, and (2) $c_i \xrightarrow{a_{i+1}}_\mathcal{T} c_{i+1}$ for all $i \in [0, n[$. In this case, we say that ρ has length n and is labelled by the word $a_1 a_2 \cdots a_n$.

Let $c, c' \in C$ and $u \in \Sigma^*$. We write $c \underset{n}{\overset{u}{\Longrightarrow}}_\mathcal{T} c'$ if one of the following two cases holds: (1) $n = 0$, $c = c'$, and $u = \epsilon$, and (2) there is a run ρ of length n from c to c' labelled by u. We also write $c \overset{u}{\Longrightarrow}_\mathcal{T}^* c'$ (resp. $c \overset{u}{\Longrightarrow}_\mathcal{T}^+ c'$) to denote that $c \underset{n}{\overset{u}{\Longrightarrow}}_\mathcal{T} c'$ for some $n \geq 0$ (resp. $n > 0$).

For every $C_1, C_2 \subseteq C$, let $Traces_\mathcal{T}(C_1, C_2) = \{u \in \Sigma^* \mid \exists (c_1, c_2) \in C_1 \times C_2, c_1 \overset{u}{\Longrightarrow}_\mathcal{T}^* c_2\}$ be the set of sequences of actions generated by the runs of $\mathcal{T}$ from a configuration in C_1 to a configuration in C_2.

For every $C' \subseteq C$, let $Pre_\mathcal{T}(C') = \{c \in C \mid \exists (c', a) \in C' \times \Sigma_\epsilon, c \xrightarrow{a}_\mathcal{T} c'\}$ be the set of immediate predecessors of C'. Let $Pre_\mathcal{T}^*$ be the reflexive-transitive closure of $Pre_\mathcal{T}$, and let $Pre_\mathcal{T}^+ = Pre_\mathcal{T} \circ Pre_\mathcal{T}^*$ where the operator $\circ$ stands for the function composition.

Finite state automata: A finite state automaton (FSA) is a tuple $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ where: (1) Q is the finite non-empty set of states, (2) Σ is the finite input alphabet, (3) $\Delta \subseteq (Q \times \Sigma_\epsilon \times Q)$ is the transition relation, (4) $I \subseteq Q$ is the set of initial states, and (5) $F \subseteq Q$ is the set of final states. We represent a transition (q, a, q') in Δ by $q \xrightarrow{a}_\mathcal{A} q'$. Moreover, if I' and F' are two subsets of Q, then we use $\mathcal{A}(I', F')$ to denote the finite state automaton defined by the tuple $(Q, \Sigma, \Delta, I', F')$.

The size of $\mathcal{A}$ is defined by $|\mathcal{A}| = (|Q| + |\Sigma| + |\Delta|)$. We use $\mathcal{T}(\mathcal{A}) = (Q, \Sigma, \Delta)$ to denote the transition system associated with $\mathcal{A}$. The language accepted (or recognized) by $\mathcal{A}$ is given by $L(\mathcal{A}) = Traces_{\mathcal{T}(\mathcal{A})}(I, F)$.

2. Generalized pushdown automata

In this section, we introduce the class of generalized pushdown automata where transitions on stacks are (1) pop the top symbol of the stack, and (2) push a word in some (effectively) given set of words L over the stack alphabet. A transition t is of the form $\delta(p, \gamma, a, p') = L$ where L is a (possibly infinite) set of words. Being in a configuration (q, w) where q is

a state and w is a stack content, t can be applied if both $p = q$ and the content of the stack is of the form $\gamma w'$ for some w'. Taking the transition and reading the input letter a (which may be the empty word), the automaton moves to the successor configuration (p', uw') where $u \in L$ (i.e., the new state is p', and γ is replaced with a word u belonging to the language L). Formally, we have:

Definition 2.1 (Generalized pushdown automata). A generalized pushdown automaton (GPA for short) is a tuple $\mathcal{P} = (P, \Sigma, \Gamma, \delta, p_0, \gamma_0, F)$ where: (1) P is the finite non-empty set of states, (2) Σ is the input alphabet, (3) Γ is the stack alphabet, (4) $\delta : P \times \Gamma \times \Sigma_\epsilon \times P \to Lang(\Gamma)$ is the transition function, (5) $p_0 \in P$ is the initial state, (6) $\gamma_0 \in \Gamma$ is the initial stack symbol, and (7) $F \subseteq P$ is the set of final states.

Next, we define the effectiveness property for generalized pushdown automata. Intuitively, the generalized pushdown automaton $\mathcal{P}$ is said to be effective if for any possible pushed language L by $\mathcal{P}$ (i.e., $\delta(p, \gamma, a, p') = L$ for some $p, p' \in Q$, $\gamma \in \Gamma$, and $a \in \Sigma_\epsilon$), the problem of checking the non-emptiness of the intersection of L and any given regular language (i.e. accepted by a finite-state automaton) is decidable.

Definition 2.2 (Effectiveness Property). A GPA $\mathcal{P} = (P, \Sigma, \Gamma, \delta, p_0, \gamma_0, F)$ is effective if and only if for every finite state automaton $\mathcal{A}$ over the alphabet Γ, it is decidable whether $L(\mathcal{A}) \cap \delta(p, \gamma, a, p') \neq \emptyset$ for all $p, p' \in P$, $\gamma \in \Gamma$, and $a \in \Sigma_\epsilon$.

A configuration of a GPA $\mathcal{P} = (P, \Sigma, \Gamma, \delta, p_0, \gamma_0, F)$ is a pair (p, w) where $p \in P$ and $w \in \Gamma^*$. The set of all configurations of $\mathcal{P}$ is denoted by $Conf(\mathcal{P})$. Similarly to the case of pushdown automata [BEM97], we use the class of $\mathcal{P}$-automata as finite symbolic representation of a set of configurations of GPA. Formally, a $\mathcal{P}$-automaton is a FSA $\mathcal{A} = (Q_\mathcal{A}, \Gamma, \Delta_\mathcal{A}, I_\mathcal{A}, F_\mathcal{A})$ such that $I_\mathcal{A} = P$. We say that a configuration (p, w) of $\mathcal{P}$ is accepted (or recognized) by $\mathcal{A}$ if $w \in L(\mathcal{A}(\{p\}, F_\mathcal{A}))$. The set of all configurations recognized by $\mathcal{A}$ is denoted by $L_\mathcal{P}(\mathcal{A})$. A set of configurations of $\mathcal{P}$ is said to be recognizable if and only if it is accepted by some $\mathcal{P}$-automaton.

$$\mathcal{P} = (\{p_0, p_1, p_2, p_f\}, \{a, b, c\}, \{\bot, \gamma_0, \gamma_1, \gamma_2\}, \delta, p_0, \bot, \{p_f\})$$

$\delta(p_0, \bot, \epsilon, p_2)$	$=$	$\{\gamma_2^i \gamma_1^i \gamma_0^i \bot \mid i \in \mathbb{N}\}$		$\delta(p_2, \gamma_2, a, p_2)$	$=$	$\{\epsilon\}$
$\delta(p_2, \gamma_1, b, p_1)$	$=$	$\{\epsilon\}$		$\delta(p_1, \gamma_1, b, p_1)$	$=$	$\{\epsilon\}$
$\delta(p_1, \gamma_0, c, p_0)$	$=$	$\{\epsilon\}$		$\delta(p_0, \gamma_0, c, p_0)$	$=$	$\{\epsilon\}$
$\delta(p_0, \bot, \epsilon, p_f)$	$=$	$\{\epsilon\}$		otherwise		$\emptyset$

Table 1: A GPA $\mathcal{P}$ for $\{\epsilon\} \cup \{a^{i_1} b^{i_1} c^{i_1} a^{i_2} b^{i_2} c^{i_2} \cdots a^{i_k} b^{i_k} c^{i_k} \mid k \geq 1 \text{ and } i_1, \ldots, i_k > 0\}$

The transition system $\mathcal{T}(\mathcal{P})$ associated with the generalized pushdown automaton $\mathcal{P}$ is defined by the tuple $(Conf(\mathcal{P}), \Sigma, \to)$ where $\to$ is the smallest transition relation such that: For every $p, p' \in P$, $\gamma \in \Gamma$, and $a \in \Sigma_\epsilon$, if $\delta(p, \gamma, a, p') \neq \emptyset$, then $(p, \gamma w) \xrightarrow{a}_{\mathcal{T}(\mathcal{P})} (p', uw)$ for all $u \in \delta(p, \gamma, a, p')$ and $w \in \Gamma^*$. Let $L(\mathcal{P}) = Traces_{\mathcal{T}(\mathcal{P})}(\{(p_0, \gamma_0)\}, F \times \{\epsilon\})$ denote the language accepted by $\mathcal{P}$.

Observe that pushdown automata can be seen as a particular class of effective GPA where $\delta(p, \gamma, a, p')$ is a finite set of words for all (p, γ, a, p').

Table 1 shows an example of an effective generalized pushdown automaton where the pushed language $\{\gamma_2^i \gamma_1^i \gamma_0^i \bot \mid i \in \mathbb{N}\}$ can be accepted by a Petri net (with reachability as acceptance condition).

2.1. **Computing the set of predecessors for an GPA.** In this section, we show that the set of predecessors of a recognizable set of configurations of an effective GPA is recognizable and effectively constructible. This is done by adapting the construction given in [BEM97, EHRS00, Sch02]. On the other hand, it is easy to observe that the set of successors of a recognizable set of configurations of an effective GPA is not recognizable in general (see the example given in Table 1).

Theorem 2.3. *For every effective generalized pushdown automaton $\mathcal{P}$, and every $\mathcal{P}$-automaton $\mathcal{A}$, it is possible to construct a $\mathcal{P}$-automaton recognizing $Pre^*_{\mathcal{T}(\mathcal{P})}(L_{\mathcal{P}}(\mathcal{A}))$.*

The rest of this section is devoted to the proof of Theorem 2.3. For that, let $\mathcal{P} = (P, \Sigma, \Gamma, \delta, p_0, \gamma_0, F)$ be an effective generalized pushdown automata and $\mathcal{A} = (Q_{\mathcal{A}}, \Gamma, \Delta_{\mathcal{A}}, I_{\mathcal{A}}, F_{\mathcal{A}})$ be an $\mathcal{P}$-automaton. Without loss of generality, we assume that $\mathcal{A}$ has no transition leading to an initial state. We compute $Pre^*_{\mathcal{T}(\mathcal{P})}(L_{\mathcal{P}}(\mathcal{A}))$ as the set of configurations recognized by an $\mathcal{P}$-automaton $\mathcal{A}_{pre^*} = (Q_{\mathcal{A}}, \Gamma, \Delta_{pre^*}, I_{\mathcal{A}}, F_{\mathcal{A}})$ obtained from $\mathcal{A}$ by means of a saturation procedure. Initially, we have $\mathcal{A}_{pre^*} = \mathcal{A}$. Then, the procedure adds new transitions to $\mathcal{A}_{pre^*}$, but no new states. New transitions are added according to the following saturation rule:

> For every $p, p' \in P$, $\gamma \in \Gamma$, and $a \in \Sigma_\epsilon$, if $\delta(p, \gamma, a, p') \neq \emptyset$, then for every $q \in Q_{\mathcal{A}}$ such that $\delta(p, \gamma, a, p') \cap L(\mathcal{A}_{pre^*}(\{p'\}, \{q\})) \neq \emptyset$, add the transition (p, γ, q) to $\mathcal{A}_{pre^*}$

It is easy to see that the saturation procedure eventually reaches a fixed point because the number of possible new transitions is finite. Moreover, the saturation procedure is well defined since the emptiness problem of the language $(\delta(p, \gamma, a, p') \cap L(\mathcal{A}_{pre^*}(\{p'\}, \{q\})))$ is decidable ($\mathcal{P}$ is an effective GPA). Then, the relation between the set of configurations recognized by $\mathcal{A}_{pre^*}$ and the set $Pre^*_{\mathcal{T}(\mathcal{P})}(L_{\mathcal{P}}(\mathcal{A}))$ is established by Lemma 2.4. (Observe that Theorem 2.3 follows from Lemma 2.4.)

Lemma 2.4. $L_{\mathcal{P}}(\mathcal{A}_{pre^*}) = Pre^*_{\mathcal{T}(\mathcal{P})}(L_{\mathcal{P}}(\mathcal{A}))$.

Lemma 2.4 is an immediate consequence of Lemma 2.5 and Lemma 2.6: Lemma 2.5 shows that $Pre^*_{\mathcal{T}(\mathcal{P})}(L_{\mathcal{P}}(\mathcal{A})) \subseteq L_{\mathcal{P}}(\mathcal{A}_{pre^*})$ while Lemma 2.6 establishes $L_{\mathcal{P}}(\mathcal{A}_{pre^*}) \subseteq Pre^*_{\mathcal{T}(\mathcal{P})}(L_{\mathcal{P}}(\mathcal{A}))$.

Lemma 2.5. *For every configuration $(p', w') \in L_{\mathcal{P}}(\mathcal{A})$, if $(p, w) \overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{P})} (p', w')$ for some $\tau \in \Sigma^*$, then $(p, w) \in L_{\mathcal{P}}(\mathcal{A}_{pre^*})$.*

Proof. Assume $(p, w) \overset{\tau}{\underset{n}{\Longrightarrow}}{}_{\mathcal{T}(\mathcal{P})} (p', w')$. We proceed by induction on n.

Basis. $n = 0$. Then, $p = p'$ and $w' = w$. Since $(p', w') \in L_{\mathcal{P}}(\mathcal{A})$ and $L_{\mathcal{P}}(\mathcal{A}) \subseteq L_{\mathcal{P}}(\mathcal{A}_{pre^*})$, we have $(p, w) \in L_{\mathcal{P}}(\mathcal{A}_{pre^*})$.

Step. $n > 0$. Then, there is a configuration $(p'', w'') \in Conf(\mathcal{P})$ such that:

$$(p, w) \overset{a}{\longrightarrow}_{\mathcal{T}(\mathcal{P})} (p'', w'') \overset{\tau'}{\underset{n-1}{\Longrightarrow}}_{\mathcal{T}(\mathcal{P})} (p', w')$$

for some $a \in \Sigma_\epsilon$ and $\tau' \in \Sigma^*$ such that $\tau = a\tau'$.

We apply the induction hypothesis to $(p'', w'') \overset{\tau}{\underset{n-1}{\Longrightarrow}}_{\mathcal{T}(\mathcal{P})} (p', w')$, and we obtain:

$$(p'', w'') \in L_{\mathcal{P}}(\mathcal{A}_{pre^*})$$

Since $(p, w) \overset{a}{\longrightarrow}_{\mathcal{T}(\mathcal{P})} (p'', w'')$, there are $\gamma \in \Gamma$ and $u, v \in \Gamma^*$ such that:

$$w = \gamma v, \ w'' = uv, \text{ and } u \in \delta(p, \gamma, a, p'')$$

Let q be a state of $\mathcal{A}_{pre^*}$ such that:

$$u \in L(\mathcal{A}_{pre^*}(\{p''\}, \{q\})) \text{ and } v \in L(\mathcal{A}_{pre^*}(\{q\}, F_{\mathcal{A}})).$$

Such a state q exists since $uv \in L(\mathcal{A}_{pre^*}(\{p''\}, F_{\mathcal{A}}))$. By the saturation rule, we have that (p, γ, q) is a transition of $\mathcal{A}_{pre^*}$ since $u \in L(\mathcal{A}_{pre^*}(\{p''\}, \{q\})) \cap \delta(p, \gamma, a, p'')$. This implies that $w = \gamma v \in L(\mathcal{A}_{pre^*}(\{p\}, F_{\mathcal{A}}))$ since $(p, \gamma, q) \in \Delta_{\mathcal{A}_{pre^*}}$ and $v \in L(\mathcal{A}_{pre^*}(\{q\}, F_{\mathcal{A}}))$. Hence, we have $(p, w) \in L_{\mathcal{P}}(\mathcal{A}_{pre^*})$. $\qquad\square$

In the following, we establish that $L_{\mathcal{P}}(\mathcal{A}_{pre^*}) \subseteq Pre^*_{\mathcal{T}(\mathcal{P})}(L_{\mathcal{P}}(\mathcal{A}))$. This is an an immediate corollary of the following lemma:

Lemma 2.6. *If $w \in L(\mathcal{A}_{pre^*}(\{p\}, \{q\}))$, then $(p, w) \overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{P})} (p', w')$ for a configuration (p', w') and $\tau \in \Sigma^*$ such that $w' \in L(\mathcal{A}_0(\{p'\}, \{q\}))$. Moreover, if q is an initial state of $\mathcal{A}_{pre^*}$, then we have $p' = q$ and $w' = \epsilon$.*

Proof. Let $\mathcal{A}_n = (Q_{\mathcal{A}}, \Gamma, \Delta_i, I_{\mathcal{A}}, F_{\mathcal{A}})$ be the $\mathcal{P}$-automaton obtained after adding n transitions to $\mathcal{A}$. In particular, we have $\mathcal{A}_0 = \mathcal{A}$. Then, it is easy to see that $L_{\mathcal{P}}(\mathcal{A}) = L_{\mathcal{P}}(\mathcal{A}_0) \subseteq L_{\mathcal{P}}(\mathcal{A}_1) \subseteq L_{\mathcal{P}}(\mathcal{A}_2) \subseteq \cdots \subseteq L_{\mathcal{P}}(\mathcal{A}_{pre^*})$.

Let n be an index such that $w \in L(\mathcal{A}_n(p, q))$ holds. We shall prove the first part of Lemma 2.6 by induction on n. The second part follows immediately from the fact that initial states have no incoming transitions in $\mathcal{A}_0$.

Basis. $n = 0$. Since $w \in L(\mathcal{A}_n(\{p\}, \{q\}))$ holds, take $w' = w$ and $p' = p$.

Step. $n > 0$. Let $t = (p'', \gamma, q')$ be the n-th transition added to $\mathcal{A}_{pre^*}$. Let m be the number of times that t is used in $p \overset{w}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}_n)} q$.

The proof is by induction on m. If $m = 0$, then we have $w \in L(\mathcal{A}_{n-1}(\{p\}, \{q\}))$, and the property (1) follows from the induction hypothesis (induction on n). So, assume that $m > 0$. Then there exist u and v such that $w = u\gamma v$ and

$$u \in L(\mathcal{A}_{n-1}(\{p\}, \{p''\})), \ p'' \overset{\gamma}{\longrightarrow}_{\mathcal{T}(\mathcal{A}_n)} q', \text{ and } v \in L(\mathcal{A}_n(\{q'\}, \{q\})).$$

The application of the induction hypothesis to $u \in L(\mathcal{A}_{n-1}(\{p\}, \{p''\}))$ yields to that:

$$(p, u) \overset{\tau'}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{P})} (p'', \epsilon) \text{ for some } \tau' \in \Sigma^*.$$

Since the transition (p'', γ, q') has been added by applying the saturation procedure, there exist p''', w_2, and $a \in \Sigma_\epsilon$ such that:

$$w_2 \in \delta(p'', \gamma, a, p'''), \text{ and } w_2 \in L(\mathcal{A}_{n-1}(\{p'''\}, \{q'\})).$$

From $w_2 \in L(\mathcal{A}_{n-1}(\{p'''\}, \{q'\}))$ and $v \in L(\mathcal{A}_n(\{q'\}, \{q\}))$, we get that there is a computation $\rho = p''' \overset{w_2v}{\underset{\mathcal{T}(\mathcal{A}_n)}{\Longrightarrow}}^* q$ such that the number of times transition t is used is strictly less than m. So, we can apply the induction hypothesis (induction on m) to ρ, and we obtain:

$(p''', w_2v) \overset{\tau''}{\underset{\mathcal{T}(\mathcal{P})}{\Longrightarrow}}^* (p', w')$ for a configuration (p', w') and $\tau'' \in \Sigma^*$ s.t. $w' \in L(\mathcal{A}_0(\{p'\}, \{q\}))$.

Putting all previous equations together, we get $w' \in L(\mathcal{A}_0(\{p'\}, \{q\}))$, and:

$$(p, w) = (p, u\gamma v) \overset{\tau'}{\underset{\mathcal{T}(\mathcal{P})}{\Longrightarrow}}^* (p'', \gamma v) \overset{a}{\underset{\mathcal{T}(\mathcal{P})}{\longrightarrow}} (p''', w_2v) \overset{\tau''}{\underset{\mathcal{T}(\mathcal{P})}{\Longrightarrow}}^* (p', w')$$

This terminates the proof of Lemma 2.6. $\square$

2.2. **Emptiness problem and closure properties for GPA.** In this section, we show that the emptiness problem is decidable for effective generalized pushdown automata. This is an immediate consequence of the fact that the set of predecessors of a recognizable set of configurations of an effective generalized pushdown automaton is also recognizable and effectively constructible.

Theorem 2.7. *The emptiness problem is decidable for effective generalized pushdown automata.*

Proof. Let $\mathcal{P} = (P, \Sigma, \Gamma, \delta, p_0, \gamma_0, F)$ be an effective generalized pushdown automaton. It is easy to see that $L(\mathcal{P}) \neq \emptyset$ if and only if $(p_0, \gamma_0) \in Pre^*_{\mathcal{T}(\mathcal{P})}(F \times \{\epsilon\})$. By Theorem 2.3, we can construct a $\mathcal{P}$-automaton $\mathcal{A}_{pre^*}$ that recognizes exactly the set $Pre^*_{\mathcal{T}(\mathcal{P})}(F \times \{\epsilon\})$ since $F \times \{\epsilon\}$ is a recognizable set of configurations. Hence, the emptiness problem for $\mathcal{P}$ is decidable since checking whether (p_0, γ_0) is in $L_\mathcal{P}(\mathcal{A}_{pre^*})$ is decidable. $\square$

Next, we show some closure properties for effective generalized pushdown automata.

Theorem 2.8. *The class of effective GPAs is closed under concatenation, union, Kleene star, projection, homomorphism, and intersection with a regular language. However, effective GPAs are not closed under intersection.*

Proof. Showing the closure of the class of effective generalized pushdown automata under concatenation, union, Kleene star, projection, homomorphism, and intersection with a regular language is similar to the case of pushdown automata. The only issue to prove is the closure under intersection. For that, let us assume by contradiction that the class of effective GPAs is closed under the intersection operation. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ two pushdown automata. Since the class of pushdown automata is a particular class of effective generalized pushdown automata and the class of effective generalized pushdown automata is closed under the intersection operation (from the contradiction's hypothesis), there is an effective generalized pushdown automaton $\mathcal{P}$ such that $L(\mathcal{P}) = L(\mathcal{P}_1) \cap L(\mathcal{P}_2)$. Applying Theorem 2.7 to $\mathcal{P}$, we obtain the decidability of the emptiness problem of the intersection of two context-free languages, which is a contradiction. $\square$

3. Ordered multi-pushdown automata

In this section, we first recall the definition of *multi-pushdown automata*. Then *ordered multi-pushdown automata* appear as a special case of multi-pushdown automata.

3.1. **Multi-pushdown automata.** Multi-pushdown automata have one read-only left to right input tape and $n \geq 1$ read-write memory tapes (stacks) with a last-in-first-out rewriting policy. A transition is of the form $t = \langle q, \gamma_1, \ldots, \gamma_n \rangle \xrightarrow{a} \langle q', \alpha_1, \ldots, \alpha_n \rangle$. Being in a configuration $(p, w_1, \ldots, w_n)$, which is composed of a state p and a stack content w_i for each stack i, t can be applied if both $q = p$ and the i-th stack is of the form $\gamma_i w_i'$ for some w_i'. Taking the transition and reading the input symbol a (which might be the empty word), the automaton moves to the successor configuration $(q', \alpha_1 w_1', \ldots, \alpha_n w_n')$.

Definition 3.1 (Multi-pushdown automata). A *multi-pushdown automaton* (MPA) is a tuple $\mathcal{M} = (n, Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0, F)$ where:
- $n \geq 1$ is the number of stacks.
- Q is the finite non-empty set of *states*.
- Σ is the finite set of *input symbols*.
- Γ is the finite set of *stack symbols* containing the special stack symbol $\perp$.
- $\Delta \subseteq \left(Q \times (\Gamma_\epsilon)^n\right) \times \Sigma_\epsilon \times \left(Q \times (\Gamma^*)^n\right)$ is the *transition relation* such that, for all $((q, \gamma_1, \ldots, \gamma_n), a, (q', \alpha_1, \ldots, \alpha_n)) \in \Delta$ and $i \in [1, n]$, we have:
  - $|\alpha_i| \leq 2$.
  - If $\gamma_i \neq \perp$, then $\alpha_i \in (\Gamma \setminus \{\perp\})^*$.
  - If $\gamma_i = \perp$, then $\alpha_i = \alpha_i' \perp$ for some $\alpha_i' \in (\Gamma_\epsilon \setminus \{\perp\})$.
- $q_0 \in Q$ is the *initial state*.
- $\gamma_0 \in (\Gamma \setminus \{\perp\})$ is the *initial stack symbol*.
- $F \subseteq Q$ is the set of *final states*.

The size of $\mathcal{M}$, denoted by $|\mathcal{M}|$, is defined by $(n + |Q| + |\Sigma| + |\Gamma| + |\Delta|)$. In the rest of this paper, we use $\langle q, \gamma_1, \ldots, \gamma_n \rangle \xrightarrow{a}_{\mathcal{M}} \langle q', \alpha_1, \ldots, \alpha_n \rangle$ to denote that the transition $((q, \gamma_1, \ldots, \gamma_n), a, (q', \alpha_1, \ldots, \alpha_n))$ is in Δ. Moreover, we denote by $\mathcal{M}(q, \gamma, q')$ the multi-pushdown automaton defined by the tuple $(n, Q, \Sigma, \Gamma, \Delta, q, \gamma, \{q'\})$.

A stack content of $\mathcal{M}$ is an element of $Stack(\mathcal{M}) = (\Gamma \setminus \{\perp\})^* \{\perp\}$. A configuration of $\mathcal{M}$ is a $(n+1)$-tuple $(q, w_1, \ldots, w_n)$ with $q \in Q$, and $w_1, \ldots, w_n \in Stack(\mathcal{M})$. A configuration $(q, w_1, \ldots, w_n)$ is final if $q \in F$ and $w_1 = \cdots = w_n = \perp$. The set of configurations of $\mathcal{M}$ is denoted by $Conf(\mathcal{M})$.

The behavior of $\mathcal{M}$ is described by its corresponding transition system $\mathcal{T}(\mathcal{M})$ defined by the tuple $(Conf(\mathcal{M}), \Sigma, \to)$ where $\to$ is the smallest transition relation satisfying the following condition: if $\langle q, \gamma_1, \ldots, \gamma_n \rangle \xrightarrow{a}_{\mathcal{M}} \langle q', \alpha_1, \ldots, \alpha_n \rangle$, then $(q, \gamma_1 w_1, \ldots, \gamma_n w_n) \xrightarrow{a}_{\mathcal{T}(\mathcal{M})} (q', \alpha_1 w_1, \ldots, \alpha_n w_n)$ for all $w_1, \ldots, w_n \in \Gamma^*$ such that $\gamma_1 w_1, \ldots, \gamma_n w_n \in Stack(\mathcal{M})$. Observe that the symbol $\perp$ marks the bottom of a stack. According to the transition relation, $\perp$ can never be popped.

The language accepted (or recognized) by $\mathcal{M}$ is defined by the set $L(\mathcal{M}) = \{\tau \in \Sigma^* \mid (q_0, \gamma_0 \perp, \perp, \ldots, \perp) \xRightarrow{\tau}^*_{\mathcal{T}(\mathcal{M})} c$ for some final configuration $c\}$.

3.2. **Symbolic representation of MPA configurations.** We show in this section how we can symbolically represent infinite sets of multi-pushdown automaton configurations using special kind of finite automata which were introduced in [Set10]. Let $\mathcal{M} = (n, Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0, F)$ be a multi-pushdown automaton. An $\mathcal{M}$-automaton for accepting configurations of $\mathcal{M}$ is a finite state automaton $\mathcal{A} = (Q_{\mathcal{M}}, \Gamma, \Delta_{\mathcal{M}}, I_{\mathcal{M}}, F_{\mathcal{M}})$ such that $I_{\mathcal{M}} = Q$. We say that a configuration $(q, w_1, \ldots, w_n)$ of $\mathcal{M}$ is accepted (or recognized) by $\mathcal{A}$ if and only if the word $w = w_1 w_2 \cdots w_n$ is in $L(\mathcal{A}(\{q\}, F_{\mathcal{M}}))$. (Notice that for every word $w \in L(\mathcal{A}(\{q\}, F_{\mathcal{M}}))$ there are unique words $w_1, \ldots, w_n \in Stack(\mathcal{M})$ such that $w = w_1 \cdots w_n$.) The set of all configurations recognized by $\mathcal{A}$ is denoted by $L_{\mathcal{M}}(\mathcal{A})$. A set of configurations of $\mathcal{M}$ is said to be recognizable if and only if it is accepted by some $\mathcal{M}$-automaton. Finally, it is easy to see that the class of $\mathcal{M}$-automata is closed under all the boolean operations and that emptiness and membership problems are decidable in polynomial time.

3.3. **Ordered multi-pushdown automata.** An ordered multi-pushdown automaton is a multi-pushdown automaton in which one can pop only from the first non-empty stack (i.e., all preceding stacks are equal to $\perp$).

Definition 3.2 (Ordered multi-pushdown automata)**.** An ordered multi-pushdown automaton (OMPA for short) is a multi-pushdown automaton $(n, Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0, F)$ where, for each transition $\langle q, \gamma_1, \ldots, \gamma_n \rangle \xrightarrow{a}_{\mathcal{M}} \langle q', \alpha_1, \ldots, \alpha_n \rangle$, there is an $i \in [1, n]$ such that $\gamma_1 = \cdots = \gamma_{i-1} = \perp$, $\gamma_i \in \Gamma_\epsilon$, and $\gamma_{i+1} = \cdots = \gamma_n = \epsilon$.

We introduce the following abbreviations: (1) For $n \geq 1$, we call an MPA/OMPA an n-MPA/n-OMPA, respectively, if its number of stacks is n, and (2) An MPA *over* Σ is an MPA with input alphabet Σ.

In the following, we consider only ordered multi-pushdown automata in some normal form. This normal form is used only to simplify the presentation. (Observe that this form is slightly more general than the one considered in [ABH08].) In such normal form, any transition, that pops a symbol from the ith stack with $i \in [2, n]$, is only allowed to push a symbol on the first stack. Furthermore, pushing symbols on the stacks from 1 to n is only allowed while popping a symbol from the first stack.

Definition 3.3. An OMPA $(n, Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0, F)$ is in normal form if Δ contains only the following types of transitions:

- $\langle q, \gamma, \epsilon, \ldots, \epsilon \rangle \xrightarrow{a}_{\mathcal{M}} \langle q', \alpha_1, \ldots, \alpha_n \rangle$ for some $q, q' \in Q$, $\gamma \in \Gamma$, $a \in \Sigma_\epsilon$, $\alpha_1 \in \Gamma^*$ and $\alpha_j \in (\Gamma_\epsilon \setminus \{\perp\})$ for all $j \in [2, n]$. This transition pops a symbol from the first stack while pushing at most two symbols on the first stack and at most one symbol on the stacks from 2 to n.

- $\langle q, \perp, \ldots, \perp, \gamma, \epsilon, \ldots, \epsilon \rangle \xrightarrow{a}_{\mathcal{M}} \langle q', \gamma' \perp, \perp, \ldots, \perp, \epsilon, \epsilon, \ldots, \epsilon \rangle$ for some $q, q' \in Q$, $\gamma, \gamma' \in (\Gamma \setminus \{\perp\})$ and $a \in \Sigma_\epsilon$. This transition pops the stack symbol γ from one of the stacks from 2 to n and pushes the stack symbol γ' on the first stack.

We can show the equivalence (with respect to language acceptance) between the class of OMPA and OMPA in the normal form.

Lemma 3.4. *An n-OMPA $\mathcal{M}$ can be transformed into an n-OMPA $\mathcal{M}'$ in normal form with linear blowup in its size such that $L(\mathcal{M}) = L(\mathcal{M}')$.*

Proof. An easy generalization of the proof for the Chomsky normal form for context-free grammars. □

In the rest of the paper, we assume that any OMPA is in the normal form. Next, we recall some properties of the class of languages recognized by n-OMPA.

Lemma 3.5 ([BCCC96]). *If $\mathcal{M}_1$ and $\mathcal{M}_2$ are two n-OMPAs over an alphabet Σ, then it is possible to construct an n-OMPA $\mathcal{M}$ over Σ such that:* (1) $L(\mathcal{M}) = L(\mathcal{M}_1) \cup L(\mathcal{M}_2)$ *and* $|\mathcal{M}| = O(|\mathcal{M}_1| + |\mathcal{M}_2|)$.

Lemma 3.6 ([BCCC96]). *Let Σ be an alphabet. Given an n-OMPA $\mathcal{M}$ over Σ and a finite state automaton $\mathcal{A}$ over Σ, then it is possible to construct an n-OMPA $\mathcal{M}'$ such that:* $L(\mathcal{M}') = L(\mathcal{M}) \cap L(\mathcal{A})$ *and* $|\mathcal{M}'| = O(|\mathcal{M}| \cdot |\mathcal{A}|)$.

4. The emptiness problem for a n-OMPA is in 2ETIME

In this section, we show that the emptiness problem for ordered pushdown automata is in 2ETIME. (We provide here a simpler proof of the 2ETIME upper bound than the one given in [ABH08].) To this aim, we show that, given an OMPA $\mathcal{M}$ with $n > 1$ stacks, it is possible to construct an effective generalized pushdown automaton $\mathcal{P}$, whose pushed languages are defined by OMPA with $(n-1)$ stacks of size $O(|\mathcal{M}|^2)$, such that the following invariant is preserved: The state and the stack content of $\mathcal{P}$ are respectively the same as the state and the content of the nth stack of $\mathcal{M}$ when its first $(n-1)$ stacks are empty (and so, $L(\mathcal{P}) \neq \emptyset$ if and only if $L(\mathcal{M}) \neq \emptyset$). Let C be a recognizable set of configurations of $\mathcal{M}$, and $Pre^*_{\mathcal{T}(\mathcal{M})}(C)$ the set of predecessors of C. Then, we can apply the saturation procedure to $\mathcal{P}$ to show that the set of configurations C_n, consisting of $Pre^*_{\mathcal{T}(\mathcal{M})}(C)$ restricted to the configurations in which the first $(n-1)$ empty stacks are empty, is recognizable and effectively constructible. To compute the intermediary configurations in $Pre^*_{\mathcal{T}(\mathcal{M})}(C)$ where the first $(n-1)$ stacks are not empty, we construct an ordered multi-pushdown automaton $\mathcal{M}'$ with $(n-1)$ stacks that: (1) performs the same transitions on its stacks as the ones performed by $\mathcal{M}$ on its first $(n-1)$ stacks, and (2) simulates a push transition of $\mathcal{M}$ over its nth stack by a transition of the finite-state automaton accepting the recognizable set of configurations C_n. Now, we can apply the induction hypothesis to $\mathcal{M}'$ and construct a finite-state automaton accepting the set of all predecessors $Pre^*_{\mathcal{T}(\mathcal{M})}(C)$.

Then, we prove, by induction on n, that the emptiness problem for the n-OMPA $\mathcal{M}$ is in 2ETIME with respect to the number of stacks. For that, we assume that the emptiness problem for $(n-1)$-OMPA can be solved in 2ETIME. This implies that the generalized pushdown automaton $\mathcal{P}$ (that simulates $\mathcal{M}$) is effective (see Definition 2.2 and Lemma 3.6). Now, we can use Theorem 2.7 to prove the decidability of the emptiness problem of the effective generalized pushdown automaton $\mathcal{P}$ (and so, of the n-OMPA $\mathcal{M}$). To show that the emptiness problem of $\mathcal{P}$ and $\mathcal{M}$ is in 2ETIME, we estimate the running time of our saturation procedure, given in section 2.1, under the assumption that the emptiness problem for $(n-1)$-OMPA can be solved in 2ETIME.

Let us give in more details of the proof described above.

4.1. **Simulation of an OMPA by an GPA.** In the following, we prove that, given an OMPA $\mathcal{M}$, we can construct a GPA $\mathcal{P}$, with transition languages defined by $(n-1)$-OMPAs of size $O(|\mathcal{M}|^2)$, such that the emptiness problem for $\mathcal{M}$ is reducible to the emptiness problem for $\mathcal{P}$. (Recall that any OMPA is assumed to be in normal form.)

Theorem 4.1. *Given an OMPA $\mathcal{M} = (n, Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0, F)$ with $n > 1$, it is possible to construct an GPA $\mathcal{P} = (P, \Sigma', \Gamma, \delta, p_0, \bot, \{p_f\})$ such that $P = Q \cup \{p_0, p_f\}$, $\Sigma' = \emptyset$, and we have:*

- *$L(\mathcal{M}) \neq \emptyset$ if and only if $L(\mathcal{P}) \neq \emptyset$, and*
- *For every $p_1, p_2 \in P$ and $\gamma \in \Gamma$, there is an $(n-1)$-OMPA $\mathcal{M}_{(p_1, \gamma, p_2)}$ over Γ such that*
$$L(\mathcal{M}_{(p_1, \gamma, p_2)}) = \big(\delta(p_1, \gamma, \epsilon, p_2)\big)^R \text{ and } |\mathcal{M}_{(p_1, \gamma, p_2)}| = O(|\mathcal{M}|^2).$$

The remaining part of this subsection is devoted to the proof of Theorem 4.1. Let us present the main steps of the construction of the generalized pushdown automaton $\mathcal{P}$. For that, let us consider an accepting run ρ of $\mathcal{M}$. This run can be seen as a sequence of runs of the form $\varsigma_1 \sigma_1 \varsigma_2 \sigma_2 \cdots \varsigma_m \sigma_m$ such that the pop transitions operations are exclusive to the first $(n-1)$-stacks (resp. the nth stack) of $\mathcal{M}$ along the sequence of runs $\varsigma_1, \varsigma_2, \ldots, \varsigma_m$ (resp. $\sigma_1, \sigma_2, \ldots, \sigma_m$). Observe that, by definition, the first $(n-1)$-stacks of $\mathcal{M}$ are empty along the runs $\sigma_1, \sigma_2, \ldots, \sigma_m$. Moreover, at the beginning of the runs $\varsigma_1, \varsigma_2, \ldots, \varsigma_m$, the OMPA $\mathcal{M}$ is in some configuration c where the first stack of $\mathcal{M}$ contains just one symbol and the stacks from 2 to $n-1$ are empty (i.e., c of the form $(q, \gamma\bot, \bot, \ldots, \bot, w)$). Observe that this is an immediate consequence of the normal form that we have considered (since $\mathcal{M}$ is only allowed to push just one symbol on the first stack while popping a symbol from the stacks from 2 to n). In the case that $\mathcal{M}$ is not in the normal form, notice that the set of all possible contents of the first $(n-1)$-stacks, at the beginning of the runs $\varsigma_1, \varsigma_2, \ldots, \varsigma_m$, is still finite. Later, we will use this observation to show how we can adapt our construction to the general case (when $\mathcal{M}$ is not in the normal form).

Then, we construct $\mathcal{P}$ such that the following invariant is preserved during the simulation of $\mathcal{M}$: The state and the content of the stack of $\mathcal{P}$ are the same as the state and the content of the nth stack of $\mathcal{M}$ when its first $(n-1)$-stacks are empty (and so, $L(\mathcal{P}) \neq \emptyset$ if and only if $L(\mathcal{M}) \neq \emptyset$). To this aim, a pushdown transition of $\mathcal{M}$ that pops a symbol γ from its nth stack is simply simulated by a pushdown transition of $\mathcal{P}$ that pops the same symbol γ. This implies that a run of the form σ_i, with $1 \leq i \leq m$, that pops the word u_i from the nth stack of $\mathcal{M}$ is simulated by a run of $\mathcal{P}$ that pops the same word u_i. Now, for every $j \in [1, m]$, we need to compute the pushed word v_j into the n-th stack of $\mathcal{M}$ during the run ς_j in order to be pushed also by $\mathcal{P}$. For that, let $L_{(q, \gamma, q')}$ be the set of all possible pushed words u into the nth stack of $\mathcal{M}$ by a run of the form $(q, \gamma\bot, \bot, \ldots, \bot, w) \overset{\tau}{\underset{\mathcal{T}(\mathcal{M})}{\Longrightarrow}}^* (q', \bot, \bot, \ldots, \bot, uw)$ where pop transitions are exclusive to the first $(n-1)$-stacks of $\mathcal{M}$. We show that this language $L_{(q, \gamma, q')}$ can be characterized by an $(n-1)$-OMPA $\mathcal{M}'(q, \gamma, q')$ over the stack alphabet of $\mathcal{M}$ that: (1) performs the same transitions on its state and $(n-1)$-stacks as the one performed by $\mathcal{M}$ on its state and its first $(n-1)$ stacks while discarding the pop transitions of $\mathcal{M}$ over the nth stack, and (2) makes visible as transition labels the pushed symbols over the nth stack of $\mathcal{M}$. Now, to simulate the run $\varsigma_j = (q, \gamma\bot, \bot, \ldots, \bot, w) \overset{\tau_j}{\underset{\mathcal{T}(\mathcal{M})}{\Longrightarrow}}^* (q', \bot, \bot, \ldots, \bot, uw)$ of $\mathcal{M}$ (which is equivalent to say that $\varsigma_j = (q, \gamma\bot, \bot, \ldots, \bot, \bot) \overset{\tau_j}{\underset{\mathcal{T}(\mathcal{M})}{\Longrightarrow}}^* (q', \bot, \bot, \ldots, \bot, u)$),

$\mathcal{P}$ can push into its stack the word u such that $u^R \in L(\mathcal{M}'(q, \gamma, q'))$. If $\mathcal{M}$ is not in the normal form, the run ς_j will be of the form $(q, \alpha_1, \ldots, \alpha_{n-1}, w) \overset{\tau_j}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q', \bot, \bot, \ldots, \bot, uw)$ (i.e., $(q, \alpha_1, \ldots, \alpha_{n-1}, \bot) \overset{\tau_j}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q', \bot, \bot, \ldots, \bot, u)$). In this case, we can construct an $(n-1)$-OMPA $M_{(q', \alpha_1, \ldots, \alpha_{n-1}, q')}$, which is precisely $\mathcal{M}'(q, \gamma, q')$ with $(q, \alpha_1, \ldots, \alpha_{n-1})$ as initial configuration, characterizing the set of all possible pushed words u on the nth-stack. Thus, the $(n-1)$-OMPA, occurring in Theorem 4.1, will be indexed by tuples of the form $(p_1, \alpha_1, \ldots, \alpha_{n-1}, p_2)$ where $p_1, p_2 \in P$ and $\alpha_1, \ldots, \alpha_{n-1} \in (\Gamma_\epsilon \cup \Gamma^2)$.

The proof of Theorem 4.1 will be structured as follows. First, we define an $(n-1)$-OMPA $\mathcal{M}'$ over the alphabet Γ that: (1) performs the same transitions on its state and $(n-1)$-stacks as the one performed by $\mathcal{M}$ on its state and its first $(n-1)$ stacks while discarding the pop transitions of $\mathcal{M}$ on the nth stack, and (2) makes visible as transition labels the pushed symbols over the nth stack of $\mathcal{M}$. Intuitively, depending on the initial and final configurations of $\mathcal{M}'$, the "language" of $\mathcal{M}'$ summarizes the effect of a sequence of pop transitions of $\mathcal{M}$ over the first $(n-1)$-stacks on the nth stack of $\mathcal{M}$. So, if we are interested only by the configurations of $\mathcal{M}$ where the first $(n-1)$ stacks are empty, a run of $\mathcal{M}$ can be seen as a sequence of alternations of a pop transition of $\mathcal{M}$ over the nth stack and a push operation over the nth stack of a word in the "language" of $\mathcal{M}'$.

Then, we construct a generalized pushdown automaton $\mathcal{P}$ such that the state and the stack content of $\mathcal{P}$ are the same as the state and the nth-stack content of $\mathcal{M}$ when the first $(n-1)$ stacks of $\mathcal{M}$ are empty. In the definition of $\mathcal{P}$, we use the $(n-1)$-OMPA $\mathcal{M}'$ to characterize the pushed word on the nth stack of $\mathcal{M}$ due to a sequence of pop transitions of $\mathcal{M}$ on the $(n-1)$ first stacks of $\mathcal{M}$. This implies that the emptiness problem for $\mathcal{M}$ is reducible to its corresponding problem for $\mathcal{P}$.

Constructing the $(n-1)$-OMPA $\mathcal{M}'$: Let us introduce the following n-OMPA $\mathcal{M}_{[1,n[} = (n, Q, \Sigma, \Gamma, \Delta_{[1,n[}, q_0, \gamma_0, F)$ such that $\Delta_{[1,n[} = \Delta \cap \left(Q \times (\Gamma_\epsilon)^{n-1} \times \{\epsilon\} \right) \times \Sigma_\epsilon \times (Q \times (\Gamma^*)^n))$. Intuitively, $\mathcal{M}_{[1,n[}$ is built up from $\mathcal{M}$ by discarding pop transitions of $\mathcal{M}$ over the nth stack. Then, let $\mathcal{M}' = (n-1, Q, \Gamma, \Gamma, \Delta', q_0, \gamma_0, F)$ be the $(n-1)$-OMPA, built out from $\mathcal{M}_{[1,n[}$, which (1) performs the same transitions on the first $(n-1)$ stacks of $\mathcal{M}_{[1,n[}$, and (2) makes visible as transition label the pushed stack symbol over the nth stack of $\mathcal{M}_{[1,n[}$. Formally, Δ' is defined as the smallest transition relation satisfying the following condition:

- If $\langle q, \gamma_1, \ldots, \gamma_{n-1}, \epsilon \rangle \overset{a}{\longrightarrow}_{\mathcal{M}_{[1,n[}} \langle q', \alpha_1, \ldots, \alpha_{n-1}, \alpha_n \rangle$ for some $q, q' \in Q$, $\gamma_1, \ldots, \gamma_{n-1} \in \Gamma_\epsilon$, $a \in \Sigma_\epsilon$, and $\alpha_1, \ldots, \alpha_n \in \Gamma^*$, then $\langle q, \gamma_1, \ldots, \gamma_{n-1} \rangle \overset{\alpha_n}{\longrightarrow}_{\mathcal{M}'} \langle q', \alpha_1, \ldots, \alpha_{n-1} \rangle$.

Observe that the pushed word α_n over the nth-stack consists in at most one symbol (i.e., $\alpha_n \in \Gamma_\epsilon$). If $\mathcal{M}$ is not in the normal formal, α_n can be of size two (say $\alpha_n = \gamma\gamma'$), and in this case we need to associate two transitions to $\mathcal{M}'$ which first read the symbol γ' and then the symbol γ.

Let us now give the relation between the effect of a sequence of operations of $\mathcal{M}_{[1,n[}$ on the nth-stack and the language of $\mathcal{M}'$.

Lemma 4.2. *For every $q, q' \in Q$, and $w_1, w'_1, \ldots, w_n, w'_n \in Stack(\mathcal{M}_{[1,n[})$, $(q, w_1, \ldots, w_n)$ $\overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q', w'_1, \ldots, w'_n)$ for some $\tau \in \Sigma^*$ if and only if there is $u \in \Gamma^*$ such that $(q, w_1, \ldots, w_{n-1}) \overset{u}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}')} (q', w'_1, \ldots, w'_{n-1})$ and $w'_n = u^R w_n$.*

Proof. To prove Lemma 4.2, it suffices to observe that the following holds: For every $q, q' \in Q$, $w_1, \ldots, w_n \in Stack(\mathcal{M}_{[1,n[})$, and $w'_1, \ldots, w'_n \in Stack(\mathcal{M}_{[1,n[})$, $(q, w_1, \ldots, w_n) \xrightarrow{a}_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q', w'_1, \ldots, w'_n)$ for some $a \in \Sigma_\epsilon$ if and only if there is $b \in \Gamma_\epsilon$ such that $(q, w_1, \ldots, w_{n-1}) \xrightarrow{b}_{\mathcal{T}(\mathcal{M}')} (q', w'_1, \ldots, w'_{n-1})$ and $w'_n = bw_n$. This observation can be established easily using the definition of $\mathcal{M}'$ and $\mathcal{M}_{[1,n[}$. $\square$

Constructing the GPA $\mathcal{P}$: We are ready now to define the generalized pushdown automaton $\mathcal{P} = (P, \Sigma', \Gamma, \delta, p_0, \perp, \{p_f\})$, with $P = Q \cup \{p_0, p_f\}$ and $\Sigma' = \emptyset$, that keeps track of the state and content of the nth stack of $\mathcal{M}$ when the first $(n-1)$ stacks are empty. Formally, $\mathcal{P}$ is built from $\mathcal{M}$ as follows: For every $p, p' \in P$ and $\gamma \in \Gamma$, we have:

- If $p = p_0$, $\gamma = \perp$, and $p' \in Q$, then $\delta(p, \gamma, \epsilon, p') = \{u^R \perp \mid u \in L(\mathcal{M}'(q_0, \gamma_0, p'))\}$.
- If $p \in F$, $\gamma = \perp$, and $p' = p_f$, then $\delta(p, \gamma, \epsilon, p') = \{\epsilon\}$.
- If $p, p' \in Q$ and $\gamma \neq \perp$ then $\delta(p, \gamma, \epsilon, p') = \bigcup_{(q,\gamma') \in \Xi} \left(L(\mathcal{M}'(q, \gamma', p'))\right)^R$ where $\Xi = \{(q, \gamma') \in (Q \times \Gamma) \mid \exists a \in \Sigma_\epsilon, \langle p, \perp, \ldots, \perp, \gamma \rangle \xrightarrow{a}_{\mathcal{M}} \langle q, \gamma'\perp, \perp, \ldots, \perp, \epsilon \rangle\}$.
- Otherwise, $\delta(p, \gamma, \epsilon, p') = \emptyset$.

Observe that for every $p_1, p_2 \in P$, and $\gamma \in \Gamma$, we can construct an $(n-1)$-OMPA $\mathcal{M}_{(p_1,\gamma,p_2)}$ over Γ such that $L(M_{(p_1,\gamma,p_2)}) = \left(\delta(p_1, \gamma, \epsilon, p_2)\right)^R$ and $|\mathcal{M}_{(p_1,\gamma,p_2)}| = O(|\mathcal{M}|^2)$. This can be easily proved using Lemma 3.5.

To complete the proof of Theorem 4.1, it remains to show that the emptiness problem for $\mathcal{M}$ is reducible to its corresponding problem for $\mathcal{P}$. This is stated by Lemma 4.3.

Lemma 4.3. $L(\mathcal{M}) \neq \emptyset$ *if and only if* $L(\mathcal{P}) \neq \emptyset$.

Proof. To prove that $L(\mathcal{M}) \neq \emptyset$ iff $L(\mathcal{P}) \neq \emptyset$, we will show that the following invariant is preserved: The state and content of $\mathcal{P}$ are the same as the state and content of the last stack of $\mathcal{M}$ when its first $(n-1)$-stacks are empty. Thus, we will split the run of $\mathcal{M}$ at the transitions that pop some symbol from the last stack. This implies that a run of $\mathcal{M}$ can be decomposed as follows: (1) First a run from the initial configuration to the first reachable configuration where the first $(n-1)$-stacks are empty, and (2) a sequence of runs that are starting from a configuration where the first $(n-1)$-stacks are empty to the first reachable configuration with empty first $(n-1)$-stacks. Lemma 4.4 and Lemma 4.5 establish the relation between these two kind of runs of $\mathcal{M}$ and the runs of $\mathcal{P}$. Lemma 4.4 shows that $\mathcal{M}$ can move from a configuration of the form $(q, \perp, \ldots, \perp, w)$ to a configuration of the form $(q', \perp, \ldots, \perp, w')$ if and only if $\mathcal{P}$ can move from the configuration (q, w) to the configuration (q', w'). Lemma 4.5 proves that $\mathcal{P}$ can move, in one step, from the initial configuration $(p_0, \perp)$ to the configuration (q, w) if and only if $\mathcal{M}_{[1,n[}$ can move from the initial configuration $(q_0, \gamma_0\perp, \perp, \ldots, \perp)$ to the configuration $(q, \perp, \perp, \ldots, \perp, w)$. As an immediate consequence of Lemma 4.4 and Lemma 4.5, we obtain that $L(\mathcal{M}) \neq \emptyset$ if and only if $L(\mathcal{P}) \neq \emptyset$.

Lemma 4.4. *For every* $q, q' \in Q$ *and* $w, w' \in Stack(\mathcal{M})$, $(q, w) \overset{\epsilon}{\Longrightarrow}^*_{\mathcal{T}(\mathcal{P})} (q', w')$ *if and only if* $(q, \perp, \ldots, \perp, w) \overset{\tau}{\Longrightarrow}^*_{\mathcal{T}(\mathcal{M})} (q', \perp, \ldots, \perp, w')$ *for some* $\tau \in \Sigma^*$.

Proof. (**The Only if direction**) In the following, we show that if $(q, w) \overset{\epsilon}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{P})} (q', w')$ then $(q, \perp, \ldots, \perp, w) \overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q', \perp, \ldots, \perp, w')$ for some $\tau \in \Sigma^*$. Assume that $(q, w) \overset{\epsilon}{\underset{i}{\Longrightarrow}}_{\mathcal{T}(\mathcal{P})} (q', w')$. We proceed by induction on i.

Basis. $i = 0$. Then $q = q'$ and $w = w'$. This implies that $(q, \perp, \ldots, \perp, w) \overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})}$ $(q', \perp, \ldots, \perp, w')$ holds with $\tau = \epsilon$.

Step. $i > 0$. Then there is a configuration $(p, v) \in Conf(\mathcal{P})$ such that:

$$(q, w) \overset{\epsilon}{\underset{i-1}{\Longrightarrow}}_{\mathcal{T}(\mathcal{P})} (p, v) \overset{\epsilon}{\rightarrow}_{\mathcal{T}(\mathcal{P})} (q', w') \tag{4.1}$$

From the definition of $\mathcal{P}$, we can show that $p \in Q$ and $v \in Stack(\mathcal{M})$ since $q, q' \in Q$ and $w, w' \in Stack(\mathcal{M})$. Thus, we can apply the induction hypothesis to $(q, w) \overset{\epsilon}{\underset{i-1}{\Longrightarrow}}_{\mathcal{T}(\mathcal{P})} (p, v)$, and we obtain:

$$(q, \perp, \ldots, \perp, w) \overset{\tau'}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (p, \perp, \ldots, \perp, v) \quad \text{for some } \tau' \in \Sigma^* \tag{4.2}$$

Since $(p, v) \overset{\epsilon}{\rightarrow}_{\mathcal{T}(\mathcal{P})} (q', w')$, there are $\gamma \in \Gamma$ and $u, v' \in \Gamma^*$ such that:

$$v = \gamma v', \quad w' = uv', \quad \text{and } u \in \delta(p, \gamma, \epsilon, q') \tag{4.3}$$

Using the definition of $\mathcal{P}$, we can show that there are $q'' \in Q$, $b \in \Sigma_\epsilon$, and $\gamma' \in \Gamma$ such that:

$$\langle p, \perp, \ldots, \perp, \gamma \rangle \overset{b}{\rightarrow}_{\mathcal{M}} \langle q'', \gamma' \perp, \ldots, \perp, \epsilon \rangle \text{ and } u^R \in L(\mathcal{M}'(q'', \gamma', q')) \tag{4.4}$$

Since $u^R \in L(\mathcal{M}'(q'', \gamma', q'))$, we obtain:

$$(q'', \gamma' \perp, \perp, \ldots, \perp) \overset{u^R}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}')} (q', \perp, \perp, \ldots, \perp) \tag{4.5}$$

Now, we can apply Lemma 4.2, to the computation of $\mathcal{M}'$ given in Equation 4.5 and the stack content v', and we obtain:

$$(q'', \gamma' \perp, \perp, \ldots, \perp, v') \overset{\tau''}{\Longrightarrow}{}^*_{\mathcal{T}(M_{[1,n[})} (q', \perp, \perp, \ldots, \perp, uv') \tag{4.6}$$

Since $\langle p, \perp, \ldots, \perp, \gamma \rangle \overset{b}{\rightarrow}_{\mathcal{M}} \langle q'', \gamma' \perp, \ldots, \perp, \epsilon \rangle$ and $v = \gamma v'$, we obtain the following computation of $\mathcal{M}$:

$$(p, \perp, \perp, \ldots, \perp, v) \overset{b}{\rightarrow}_{\mathcal{T}(\mathcal{M})} (q'', \gamma' \perp, \perp, \ldots, \perp, v') \tag{4.7}$$

Putting together Equation 4.2, Equation 4.6, and Equation 4.7, we obtain:

$$(q, \perp, \ldots, \perp, w) \overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q', \perp, \ldots, \perp, w') \text{ with } \tau = \tau' b \tau'' \tag{4.8}$$

This terminates the Only if direction of Lemma 4.4.

(**The If direction**) In the following, we show that if $(q, \perp, \ldots, \perp, w) \overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})}$ $(q', \perp, \ldots, \perp, w')$ for some $\tau \in \Sigma^*$, then $(q, w) \overset{\epsilon}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{P})} (q', w')$. Let us assume that

$\rho = (q, \bot, \ldots, \bot, w) \overset{\tau}{\underset{\mathcal{T}(\mathcal{M})}{\Longrightarrow}}{}^* (q', \bot, \ldots, \bot, w')$ for some $\tau \in \Sigma^*$. The proof is by induction on the number of times that a pop transition over the nth stack of $\mathcal{M}$ is used in the run ρ. Let m be the number of times that a transition in $\Delta_n = (\Delta \setminus \Delta_{[1,n[})$ is used in ρ.

Basis. $m = 0$. Then, $q = q'$, $w = w'$, and $\tau = \epsilon$ since no transitions from Δ_n are used in ρ, and no transitions from $\Delta_{[1,n[}$ can be applied along the run ρ. This implies that $(q, w) \overset{\epsilon}{\underset{\mathcal{T}(\mathcal{P})}{\Longrightarrow}}{}^* (q', w')$ holds.

Step. $m > 0$. Then, there are $\gamma, \gamma' \in (\Gamma \setminus \{\bot\})$, $v \in \Gamma^*$, and $q_1, q_2 \in Q$ such that:

$$\rho_1 = (q, \bot, \ldots, \bot, w) \overset{\tau'}{\underset{\mathcal{T}(\mathcal{M})}{\Longrightarrow}}{}^* (q_1, \bot, \ldots, \bot, \gamma v) \tag{4.9}$$

$$\rho_2 = (q_1, \bot, \ldots, \bot, \gamma v) \overset{a}{\underset{\mathcal{T}(\mathcal{M}_n)}{\longrightarrow}} (q_2, \gamma' \bot, \ldots, \bot, v) \tag{4.10}$$

$$\rho_3 = (q_2, \gamma' \bot, \ldots, \bot, v) \overset{\tau''}{\underset{\mathcal{T}(\mathcal{M}_{[1,n[})}{\Longrightarrow}}{}^* (q', \bot, \ldots, \bot, w') \tag{4.11}$$

for some $\tau', \tau'' \in \Sigma^*$ and $a \in \Sigma_\epsilon$ such that $\tau = \tau' a \tau''$.

Observe that such decomposition of ρ is possible since in order to apply a transition in Δ_n, the first $(n-1)$ stacks of $\mathcal{M}$ must be empty.

Now, we can apply the induction hypothesis to ρ_1, and we obtain the following run of $\mathcal{P}$:

$$(q, w) \overset{\epsilon}{\underset{\mathcal{T}(\mathcal{P})}{\Longrightarrow}}{}^* (q_1, \gamma v) \tag{4.12}$$

We can also apply Lemma 4.2 to the run ρ_3, and we obtain that there is $u \in \Gamma^*$ such that:

$$u \in L(\mathcal{M}'(q_2, \gamma', q')), \text{ and } w' = u^R v \tag{4.13}$$

From the run ρ_2, we get $\langle q_1, \bot, \bot, \ldots, \bot, \gamma \rangle \overset{a}{\underset{\mathcal{M}}{\longrightarrow}} \langle q_2, \gamma' \bot, \bot, \ldots, \bot, \epsilon \rangle$. Moreover, we have $u \in L(\mathcal{M}'(q_2, \gamma', q'))$. This implies that $u^R \in \delta(q_1, \gamma, \epsilon, q')$. This means that:

$$(q, \gamma v) \overset{\epsilon}{\underset{\mathcal{T}(\mathcal{P})}{\longrightarrow}} (q', u^R v) = (q', w') \tag{4.14}$$

Putting together Equations 4.12 and 4.14, we obtain:

$$(q, w) \overset{\epsilon}{\underset{\mathcal{T}(\mathcal{P})}{\Longrightarrow}}{}^* (q, \gamma v) \overset{\epsilon}{\underset{\mathcal{T}(\mathcal{P})}{\longrightarrow}} (q', w') \tag{4.15}$$

This terminates the If direction of Lemma 4.4. $\qquad\square$

Next, we prove that $\mathcal{P}$ can perform a transition from the initial configuration $(p_0, \bot)$ to a configuration of the form (q, w) if and only if $\mathcal{M}_{[1,n[}$ can move from the initial configuration $(q_0, \gamma_0 \bot, \bot, \ldots, \bot)$ to the configuration $(q, \bot, \bot, \ldots, \bot, w)$.

Lemma 4.5. *For every $q \in Q$ and $w \in Stack(\mathcal{M})$, $(p_0, \bot) \xrightarrow{\epsilon}_{\mathcal{T}(\mathcal{P})}(q, w)$ if and only if $(q_0, \gamma_0 \bot, \bot, \ldots, \bot) \overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q, \bot, \bot, \ldots, \bot, w)$ for some $\tau \in \Sigma^*$.*

Proof. (**The If direction**) Assume that $\rho = (q_0, \gamma_0 \bot, \bot, \ldots, \bot) \overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})}$ $(q, \bot, \bot, \ldots, \bot, w)$ for some $\tau \in \Sigma^*$. Then, we can apply Lemma 4.2 to the run ρ. This implies that there is $u \in \Gamma^*$ such that:

$$(q_0, \gamma_0, \bot, \ldots, \bot) \overset{u}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}')}(q, \bot, \ldots, \bot) \quad \text{and} \quad w = u^R \bot$$

This means that $u \in L(\mathcal{M}'(q_0, \gamma_0, q))$, and therefore $u^R \bot \in \delta(p_0, \bot, \epsilon, q)$. This implies that the system $\mathcal{T}(\mathcal{P})$ can move from the configuration $(p_0, \bot)$ to the configuration $(q, u^R \bot)$:

$$(p_0, \bot) \xrightarrow{\epsilon}_{\mathcal{T}(\mathcal{P})}(q, u^R \bot) = (q, w)$$

This terminates the proof of the If direction.

(**The Only if direction**) Assume that $(p_0, \bot) \xrightarrow{\epsilon}_{\mathcal{T}(\mathcal{P})}(q, w)$. Then, from the definition of $\mathcal{P}$, there is an $u \in \Gamma^*$ such that:

$$w = u^R \bot \quad \text{and} \quad u \in L(\mathcal{M}'(q_0, \gamma_0, q))$$

From the definition of $L(\mathcal{M}'(q_0, \gamma_0, q))$, we have $\rho = (q_0, \gamma_0 \bot, \bot, \ldots, \bot) \overset{u}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}')}$ $(q, \bot, \bot, \ldots, \bot)$. We can apply Lemma 4.2 to the run ρ, and we obtain that there is $\tau \in \Sigma^*$ such that:

$$(q_0, \gamma_0 \bot, \bot, \ldots, \bot) \overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})}(q, \bot, \ldots, \bot, u^R \bot) = (q, \bot, \ldots, \bot, w)$$

This terminates the Only if direction and the proof of Lemma 4.5. $\qquad \square$

Now, we are ready to prove that the emptiness problem for $\mathcal{M}$ is reducible to the emptiness problem for $\mathcal{P}$.

(**The If direction**) Assume that $L(\mathcal{P}) \neq \emptyset$. This implies that:

$$(p_0, \bot) \overset{\epsilon}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{P})} (p_f, \epsilon) \tag{4.16}$$

This means that there is a state $q \in F$ such that:

$$(p_0, \bot) \overset{\epsilon}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{P})} (q, \bot) \xrightarrow{\epsilon}_{\mathcal{T}(\mathcal{P})}(p_f, \epsilon) \tag{4.17}$$

From the definition of the transition function of $\mathcal{P}$, there are $q' \in Q$ and $w \in Stack(\mathcal{M})$ such that:

$$\rho_1 = (p_0, \bot) \xrightarrow{\epsilon}_{\mathcal{T}(\mathcal{P})}(q', w) \tag{4.18}$$

$$\rho_2 = (q', w) \overset{\epsilon}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{P})} (q, \bot) \tag{4.19}$$

We can apply Lemma 4.5 to the run ρ_1, and we obtain that there is $\tau' \in \Sigma^*$ such that:

$$(q_0, \gamma_0\bot, \ldots, \bot) \overset{\tau'}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q', \bot, \ldots, \bot, w) \tag{4.20}$$

We can also apply Lemma 4.4 to the run ρ_2, and we obtain that there is $\tau'' \in \Sigma^*$ such that:

$$(q', \bot, \ldots, \bot, w) \overset{\tau''}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q, \bot, \ldots, \bot) \tag{4.21}$$

Putting together Equations 4.20 and 4.21, we get:

$$(q_0, \gamma_0\bot, \ldots, \bot) \overset{\tau'}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q', \bot, \ldots, \bot, w) \overset{\tau''}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q, \bot, \ldots, \bot) \tag{4.22}$$

This shows that $L(\mathcal{M}) \neq \emptyset$ since $q \in F$, and this terminates the proof of the If direction.

(**The Only if direction**:) Assume that $L(\mathcal{M}) \neq \emptyset$. Then, there is a state $q \in F$ such that $(q_0, \gamma_0\bot, \bot, \ldots, \bot) \overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q, \bot, \ldots, \bot)$ for some $\tau \in \Sigma^*$.

This implies that there is a state $q' \in Q$, $\tau', \tau'' \in \Sigma^*$, and $w \in Stack(\mathcal{M})$ such that:

$$\rho_1 = (q_0, \gamma_0\bot, \ldots, \bot) \overset{\tau'}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q', \bot, \ldots, \bot, w) \tag{4.23}$$

$$\rho_2 = (q', \bot, \ldots, \bot, w) \overset{\tau''}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q, \bot, \ldots, \bot) \tag{4.24}$$

We can apply Lemma 4.5 to the run ρ_1, and we obtain:

$$(p_0, \bot) \overset{\epsilon}{\longrightarrow}_{\mathcal{T}(\mathcal{P})} (q', w) \tag{4.25}$$

We can also apply Lemma 4.4 to the run ρ_2, and we obtain that:

$$(q', w) \overset{\epsilon}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{P})} (q, \bot) \tag{4.26}$$

Putting together Equations 4.25 and 4.26, we get that:

$$(p_0, \bot) \overset{\epsilon}{\longrightarrow}_{\mathcal{T}(\mathcal{P})} (q', w) \overset{\epsilon}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{P})} (q, \bot) \tag{4.27}$$

Moreover, we can apply the transition function $\delta(q, \bot, \epsilon, p_f) = \epsilon$ to the configuration $(q, \bot)$, and we obtain the following computation of $\mathcal{T}(\mathcal{P})$:

$$(p_0, \bot) \overset{\epsilon}{\longrightarrow}_{\mathcal{T}(\mathcal{P})} (q', w) \overset{\epsilon}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{P})} (q, \bot) \overset{\epsilon}{\longrightarrow}_{\mathcal{T}(\mathcal{P})} (p_f, \epsilon) \tag{4.28}$$

This shows that $L(\mathcal{P}) \neq \emptyset$, and this terminates the proof of the Only if direction. $\qquad \square$

4.2. **Emptiness of a n-OMPA is in 2ETIME.** In the following, we show that the emptiness problem for a n-OMPA is in 2ETIME with respect to the number of stacks. The proof is done by induction on the number of stacks. First, we use the induction hypothesis, that the emptiness problem for OMPA with $(n-1)$-stacks is decidable, to show that the generalized pushdown automaton $\mathcal{P}$ is effective (and so the emptiness problem for $\mathcal{P}$ is decidable). Once the effectiveness property of $\mathcal{P}$ has been established, we estimate the running time of our saturation procedure for $\mathcal{P}$, given in section 2.1, under the assumption that the emptiness problem. for $(n-1)$-OMPA can be solved in 2ETIME. We show that the emptiness problem of $\mathcal{P}$ (and so $\mathcal{M}$) is in 2ETIME

Theorem 4.6. *The emptiness problem for an n-OMPA $\mathcal{M}$ can be solved in time $O(|\mathcal{M}|^{2^{dn}})$ for some constant d.*

Proof. Let $\mathcal{M} = (n, Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0, F)$ be an n-OMPA. To prove Theorem 4.6, we proceed by induction on the number of stacks n.

Basis. $n = 1$. Then, $\mathcal{M}$ is a pushdown automaton. From [BEM97], we know that the emptiness problem for $\mathcal{M}$ can be solved in polynomial time in $|\mathcal{M}|$.

Step. $n > 1$. Then, we can apply Theorem 4.1 to construct a generalized pushdown automaton $\mathcal{P} = (P, \emptyset, \Gamma, \delta, p_0, \perp, \{p_f\})$, with $P = Q \cup \{p_0, p_f\}$, such that:

- $L(\mathcal{M}) \neq \emptyset$ if and only if $L(\mathcal{P}) \neq \emptyset$, and
- For every $p_1, p_2 \in P$ and $\gamma \in \Gamma$, there is an $(n-1)$-OMPA $\mathcal{M}_{(p_1, \gamma, p_2)}$ over Γ such that
  $L(\mathcal{M}_{(p_1, \gamma, p_2)}) = \big(\delta(p_1, \gamma, \epsilon, p_2)\big)^R$ and $|\mathcal{M}_{(p_1, \gamma, p_2)}| = O(|\mathcal{M}|^2)$.

It is easy to observe that $\mathcal{P}$ is an effective generalized pushdown automaton. This is established by the following lemma.

Lemma 4.7. *$\mathcal{P}$ is an effective generalized pushdown automaton.*

Proof. To prove the effectiveness property of $\mathcal{P}$, we need to show that for every finite state automaton $\mathcal{A}$ over the alphabet Γ, the problem of checking whether $L(\mathcal{A}) \cap \delta(p_1, \gamma, \epsilon, p_2) \neq \emptyset$ is decidable for all $p_1, p_2 \in P$ and $\gamma \in \Gamma$. It can be easy shown that $L(\mathcal{A}) \cap \delta(p_1, \gamma, \epsilon, p_2) \neq \emptyset$ if and only if $L(\mathcal{A})^R \cap (\delta(p_1, \gamma, \epsilon, p_2))^R \neq \emptyset$.

Let $\mathcal{A}$ be a given finite state automaton, $p_1, p_2 \in P$ two states of $\mathcal{P}$, and $\gamma \in \Gamma$ a stack symbol of $\mathcal{P}$. Using Lemma 3.6, we can construct an $(n-1)$-OMPA $\mathcal{M}'$ such that $L(\mathcal{M}') = (L(\mathcal{A}))^R \cap L(M_{(p_1, \gamma, p_2)}) = (L(\mathcal{A}))^R \cap (\delta(p_1, \gamma, \epsilon, p_2))$ since we have $M_{(p_1, \gamma, p_2)} = (\delta(p_1, \gamma, \epsilon, p_2))^R$. Now, we can apply the induction hypothesis to $\mathcal{M}'$ to show that the checking whether $L(\mathcal{M}') \neq \emptyset$ is decidable. Thus, $\mathcal{P}$ is an effective generalized pushdown automaton. $\square$

From Theorem 2.7, Theorem 4.1, and Lemma 4.7, we deduce that the emptiness problem for the n-OMPA $\mathcal{M}$ is decidable.

Next, we will estimate the running time of the decision procedure. From Theorem 2.7, we know that the emptiness problem of $\mathcal{P}$ is reducible to compute the set of predecessors of the configuration (p_f, ϵ) since $L(\mathcal{P}) \neq \emptyset$ if and only if $(p_0, \perp) \in Pre^*_{\mathcal{T}(\mathcal{P})}(\{p_f\} \times \{\epsilon\})$.

Let $\mathcal{A}$ be the $\mathcal{P}$-automaton that recognizes the configuration (p_f, ϵ) of $\mathcal{P}$. It is easy to see that such $\mathcal{P}$-automaton $\mathcal{A}$, with $|\mathcal{A}| = O(|\mathcal{M}|)$, is effectively constructible. Now, we need to analysis the running time of the saturation procedure (given in section 2.1) applied to $\mathcal{A}$. For that, let $\mathcal{A}_0, \ldots, \mathcal{A}_i$ be the sequence of $\mathcal{P}$-automaton obtained from the saturation procedure such that $\mathcal{A}_0 = \mathcal{A}$ and $L_{\mathcal{P}}(\mathcal{A}_i) = Pre^*_{\mathcal{T}(\mathcal{P})}(L_{\mathcal{P}}(\mathcal{A}))$. Then, we have $i = O(|\mathcal{M}|^3)$ since the number of possible new transitions of $\mathcal{A}$ is finite. Moreover, at each step j, with $0 \leq j \leq i$, we need to check, for every state q of $\mathcal{A}$, $p, p' \in P$, and $\gamma \in \Gamma$, whether $L(\mathcal{A}_j)(\{p'\}, \{q\}) \cap \delta(p, \gamma, \epsilon, p') \neq \emptyset$.

Using Lemma 3.6, we can construct, in polynomial time in $|\mathcal{M}|$, an $(n-1)$-OMPA $\mathcal{M}'_{(q,p,\gamma,p')}$ such that $L(\mathcal{M}'_{(q,p,\gamma,p')}) = (L(\mathcal{A}_j)(\{p'\}, \{q\}))^R \cap L(M_{(p,\gamma,p')})$ and $|\mathcal{M}'_{(q,p,\gamma,p')}| \leq c(|\mathcal{M}|^3)$ for some constant c. Now, we can apply the induction hypothesis to $\mathcal{M}'_{(q,p,\gamma,a,p')}$, and we obtain that the problem of checking whether $L(\mathcal{M}'_{(q,p,\gamma,a,p')}) \neq \emptyset$ can be solved

in time $O\big((c\,|\mathcal{M}|^3)^{2^{d(n-1)}}\big)$. Putting together all these equations, we obtain that the problem of checking whether $(p_0, \perp) \in Pre^*_{\mathcal{T}(\mathcal{P})}(\{p_f\} \times \{\epsilon\})$ can be solved in time $O\big(|\mathcal{M}|^3|\mathcal{M}|^5(c\,|\mathcal{M}|^3)^{2^{d(n-1)}}\big)$. By taking a constant d as big as needed, we can show that the problem of checking whether $L(\mathcal{M}) \neq \emptyset$ can be solved in time $O(|\mathcal{M}|^{2^{dn}})$. $\qquad\square$

5. Computing the set of predecessors for OMPA

In this section, we show that the set of predecessors of a recognizable set C of configurations of an OMPA is recognizable and effectively constructible (see Theorem 5.6). To simplify the presentation, we can assume without loss of generality that the set C contains only one configuration of the form $(q_f, \perp, \ldots, \perp)$ where all the stacks are empty. This result is established by Lemma 5.1.

Lemma 5.1. *Let $\mathcal{M} = (n, Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0, F)$ be an OMPA and $\mathcal{A}$ be an $\mathcal{M}$-automaton. Then, it is possible to construct, in time and space polynomial in $(|\mathcal{M}| + |\mathcal{A}|)$, an OMPA $\mathcal{M}' = (n, Q' \cup \{q_f\}, \Sigma, \Gamma', \Delta', q_0, \gamma_0, F)$ where $Q \subseteq Q'$, $q_f \notin Q'$, and $|\mathcal{M}'| = O(|\mathcal{M}| \cdot |\mathcal{A}|)$ such that for every $c \in Conf(\mathcal{M})$, $c \in Pre^*_{\mathcal{T}(\mathcal{M})}(L_{\mathcal{M}}(\mathcal{A}))$ if and only if $c \in Pre^*_{\mathcal{T}(\mathcal{M}')}(\{(q_f, \perp, \ldots, \perp)\})$.*

Proof. The proof is similar to the case of standard pushdown automata. Technically, this can be done by adding to the OMPA $\mathcal{M}$ some pop transitions that check, in nondeterministic way, if the current configuration belongs to $L_{\mathcal{M}}(\mathcal{A})$ by simulating the $\mathcal{M}$-automaton $\mathcal{A}$. Let $\mathcal{A} = (Q_{\mathcal{M}}, \Gamma, \Delta_{\mathcal{M}}, I_{\mathcal{M}}, F_{\mathcal{M}})$ be the $\mathcal{M}$-automaton. We assume w.l.o.g that $\mathcal{A}$ has no transition leading to its initial states and that there is no transition of $\mathcal{A}$ labeled by the empty word.

We construct the OMPA $\mathcal{M}' = (n, Q' \cup \{q_f\}, \Sigma, \Gamma', \Delta', q_0, \gamma_0, F)$ as follows:

- $Q' = Q \cup (Q_{\mathcal{M}} \times [1, n+1])$. The set of states Q' is precisely the union of the set of states of $\mathcal{M}$ and the set of states of $\mathcal{A}$ indexed by the stack identities. (The index $n+1$ is used to mark the end of the simulation of $\mathcal{A}$ by $\mathcal{M}'$). Moreover, we assume that $\mathcal{M}'$ has a fresh state $q_f \notin Q'$.

- $\Gamma' = \Gamma \cup \{\sharp\}$ such that $\sharp \notin \Gamma$. The fresh stack symbol $\sharp$ is used to ensure that $\mathcal{M}'$ respects the constraints imposed by the normal formal. Intuitively, this symbol will be pushed on the first stack whenever a symbol is popped from a stack with an index from 2 to n, during the simulation of the $\mathcal{M}$-automaton $\mathcal{A}$, and then this symbol will be popped from the first stack.

- Δ' is the smallest transition relation such that the following conditions are satisfied:

  - **First Phase:** In this phase the OMPA $\mathcal{M}'$ behaves exactly as the OMPA $\mathcal{M}'$. This corresponds to $\Delta \subseteq \Delta'$.

  - **Second phase:** In the second phase, $\mathcal{M}'$ checks if the current configuration is accepted by $\mathcal{A}$. This is done by allowing $\mathcal{M}'$ to start, in non-deterministically way, the simulation of $\mathcal{A}$ while popping the read symbols from their corresponding stacks. Formally, we have:

* For every transition $q \xrightarrow{\gamma}_{\mathcal{A}} p$ with $q \in Q$ and $\gamma \in (\Gamma \setminus \{\bot\})$, we have $\langle q, \gamma, \epsilon, \ldots, \epsilon \rangle \xrightarrow{\epsilon}_{\mathcal{M}'} \langle (p, 1), \epsilon, \epsilon, \ldots, \epsilon \rangle$. This means that, in non-deterministically way, the checking of whether the current configuration is accepted by $\mathcal{A}$ can be started by the simulation of a transition of $\mathcal{A}$ from the current state q. This transition of $\mathcal{M}'$ corresponds to the case where the first stack is not empty

* For every $q \overset{\bot^i}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A})} p$ with $i \in [1, n]$ and $q \in Q$, we have $\langle q, \bot, \epsilon, \ldots, \epsilon \rangle \xrightarrow{\epsilon}_{\mathcal{M}'} \langle (p, i+1), \bot, \epsilon, \ldots, \epsilon \rangle$. This means that the simulation of $\mathcal{A}$ by $\mathcal{M}$ can be started and the first i stacks are empty. Observe that the state $(p, i+1)$ of $\mathcal{M}'$ corresponds to the fact that the current state of $\mathcal{A}$ is p and that we are currently checking the stack $i+1$.

* For every $i \in [1, n]$ and $p \overset{\bot^j}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A})} p'$ for some $j \in [1, n-i+1]$, we have $\langle (p, i), \bot, \epsilon, \ldots, \epsilon \rangle \xrightarrow{\epsilon}_{\mathcal{M}'} \langle (p', i+j), \bot, \epsilon, \ldots, \epsilon \rangle$. This corresponds to the simulation of a sequence of transitions of $\mathcal{A}$ that checks if the stacks from i to $(i+j-1)$ are empty. In this case, we move the current state from p to p' and we start checking the stack of index $(i+j)$.

* For every $i \in [1, n]$ and $p \xrightarrow{\gamma}_{\mathcal{A}} p'$ for some $\gamma \in (\Gamma \setminus \{\bot\})$, we have $\langle (p, i), \gamma_1, \ldots, \gamma_n \rangle \xrightarrow{\epsilon}_{\mathcal{M}'} \langle (p', i), \alpha_1, \ldots, \alpha_n \rangle$ with $\gamma_1 = \cdots = \gamma_{i-1} = \bot$, $\gamma_i = \gamma$, $\gamma_{i+1} = \cdots = \gamma_n = \epsilon$, $\alpha_1 = \sharp \cdot \bot$, $\alpha_2 = \cdots = \alpha_{i-1} = \bot$, and $\alpha_i = \cdots = \alpha_n = \epsilon$. The simulation of a transition of $\mathcal{A}$, that reads the symbol γ from the ith-stack, is performed by $\mathcal{M}'$ by a transition that pops γ form the ith-stack and pushes the fresh symbol $\sharp$ into the first stack.

* For every index $i \in [1, n+1]$ and state $p \in Q_{\mathcal{A}}$, we have $\langle (p, i), \sharp, \epsilon, \ldots, \epsilon \rangle \xrightarrow{\epsilon}_{\mathcal{M}'} \langle (p, i), \epsilon, \ldots, \epsilon \rangle$. This transition pops the fresh symbol $\sharp$ from the first stack. Recall that this fresh symbol is introduced in the only aim of ensuring the normal form of $\mathcal{M}'$.

* For every $p \in F$, we have $\langle (p, n+1), \bot, \epsilon, \ldots, \epsilon \rangle \xrightarrow{\epsilon}_{\mathcal{M}'} \langle q_f, \bot, \epsilon, \ldots, \epsilon \rangle$. This transition ends the simulation of $\mathcal{A}$ by $\mathcal{M}'$ after verifying that the current configuration of $\mathcal{M}$ is accepted by $\mathcal{A}$.

Then it is easy to see that for every $c \in Conf(\mathcal{M})$, $c \in Pre^*_{\mathcal{T}(\mathcal{M})}(L_{\mathcal{M}}(\mathcal{A}))$ if and only if $c \in Pre^*_{\mathcal{T}(\mathcal{M}')}(\{(q_f, \bot, \ldots, \bot)\})$. $\qquad \square$

In the following, we show that the set of configurations C' of the form $(q', \bot, \ldots, \bot, w')$ from which the OMPA $\mathcal{M}$ can reach a configuration of the form $(q, \bot, \ldots, \bot)$, where all the stacks are empty, is recognizable and effectively constructible.

Lemma 5.2. *Let $\mathcal{M} = (n, Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0, F)$ be an OMPA and $q \in Q$ be a state. Then, it is possible to construct, in time $O(|\mathcal{M}|^{2^{dn}})$ where d is a constant, an $\mathcal{M}$-automaton $\mathcal{A}$ such that $|\mathcal{A}| = O(|\mathcal{M}|)$ and $c \in L_{\mathcal{M}}(\mathcal{A})$ if and only if $c \in Pre^*_{\mathcal{T}(\mathcal{M})}(\{(q, \bot, \ldots, \bot)\})$ and $c = (q', \bot, \ldots, \bot, w)$ for some $q' \in Q$ and $w \in Stack(\mathcal{M})$.*

Proof. Lemma 4.4 shows that, given an OMPA $\mathcal{M}$ with n stacks, it is possible to construct an effective generalized pushdown automaton $\mathcal{P}$, whose pushed languages are defined by OMPA with $(n-1)$ stacks, such that the following invariant is preserved: The state and the stack's content of $\mathcal{P}$ are the same as the state and the content of the nth stack of $\mathcal{M}$

when its first $(n-1)$ stacks are empty. Then, we can make use of Theorem 2.3, which shows the set of all predecessors of a recognizable set of configurations is an effectively constructible recognizable set for effective generalized pushdown automata, to show that Lemma 5.2 holds.   □

Next, we state our main theorem which is a generalization of the result obtained in for bounded-phase visibly multi-stack pushdown automata [Set10].

Theorem 5.3. *Let* $\mathcal{M} = (n, Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0, F)$ *be an OMPA and* $q \in Q$ *be a state. Then, it is possible to construct, in time* $O(|\mathcal{M}|^{2^{dn}})$ *where* d *is a constant, a* $\mathcal{M}$-*automaton* $\mathcal{A}$ *such that* $|\mathcal{A}| = O(|\mathcal{M}|^{2^{dn}})$ *and* $L_{\mathcal{M}}(\mathcal{A}) = Pre^*_{\mathcal{T}(\mathcal{M})}(\{(q, \bot, \ldots, \bot)\})$.

Proof. From Lemma 5.2, we know that the set of configurations C_n, consisting of $Pre^*_{\mathcal{T}(\mathcal{M})}(\{(q, \bot, \ldots, \bot)\})$ restricted to the configurations in which the first $(n-1)$ empty stacks are empty, is recognizable and effectively constructible. To compute the intermediary configurations in $Pre^*_{\mathcal{T}(\mathcal{M})}(\{(q, \bot, \ldots, \bot)\})$ where the first $(n-1)$ stacks are not empty, we construct an ordered multi-pushdown automaton $\mathcal{M}'$ with $(n-1)$ stacks that: (1) performs the same transitions on its stacks as the ones performed by $\mathcal{M}$ on its first $(n-1)$ stacks, and (2) simulates a push transition of $\mathcal{M}$ over its nth stack by a transition of the finite-state automaton accepting the recognizable set of configurations C_n. Now, we can apply the induction hypothesis to $\mathcal{M}'$ and construct a finite-state automaton accepting the set of all predecessors $Pre^*_{\mathcal{T}(\mathcal{M})}(\{(q, \bot, \ldots, \bot)\})$.

We proceed by induction on the number of stacks of the OMPA $\mathcal{M}$.

Basis. $n = 1$. Then, $\mathcal{M}$ is a pushdown automaton. From [BEM97], we know that such an $\mathcal{M}$-automaton $\mathcal{A}$ for $\mathcal{M}$ can be constructed in polynomial time in $|\mathcal{M}|$.

Step. $n > 1$. Then, we can use Lemma 5.2 to construct, in time $O(|\mathcal{M}|^{2^{dn}})$ where d is a constant, an $\mathcal{M}$-automaton $\mathcal{A}' = (Q_{\mathcal{A}'}, \Gamma, \Delta_{\mathcal{A}'}, Q, F_{\mathcal{A}'})$ such that $|\mathcal{A}'| = O(|\mathcal{M}|)$ and $(q'', \bot, \ldots, \bot, w) \in L_{\mathcal{M}}(\mathcal{A}')$ if and only if $(q'', \bot, \ldots, \bot, w) \overset{\tau'}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q, \bot, \ldots, \bot)$ for some $\tau' \in \Sigma^*$. Afterwards, we assume without loss of generality that the $\mathcal{M}$-automaton has no ϵ-transitions.

Let $\mathcal{M}_{[1,n[} = (n, Q, \Sigma, \Gamma, \Delta_{[1,n[}, q_0, \gamma_0, F)$ be the OMPA built from $\mathcal{M}$ by discarding the set of pop transitions of $\mathcal{M}$ over the nth stack (as defined in Section 4.1). (Recall that $\Delta_{[1,n[} = \Delta \cap \big((Q \times (\Gamma_\epsilon)^{n-1} \times \{\epsilon\}) \times \Sigma_\epsilon \times (Q \times (\Gamma^*)^n)\big)$). Then, it is easy to see that for every configuration $(q', w_1, \ldots, w_n)$ in $Pre^*_{\mathcal{T}(\mathcal{M})}(\{(q, \bot, \ldots, \bot)\})$, there are $q'' \in Q$, $w \in Stack(\mathcal{M})$, and $\tau', \tau \in \Sigma^*$ such that:

$$(q', w_1, \ldots, w_n) \overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q'', \bot, \ldots, \bot, w) \overset{\tau'}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q, \bot, \ldots, \bot)$$

Since the OMPA $\mathcal{M}_{[1,n[}$ can only have push transitions over its nth stack, we have $(q', w_1, \ldots, w_n) \overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q'', \bot, \ldots, \bot, w)$ if and only if there is $v \in (\Gamma \setminus \{\bot\})^*$ such that $w = v w_n$ and $(q', w_1, \ldots, w_{n-1}, \bot) \overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q'', \bot, \ldots, \bot, v)$ (see Lemma 4.2).

Let $\mathcal{M}' = (n-1, Q \times Q_{\mathcal{A}'}, \Sigma, \Gamma, \Delta', q_0', \gamma_0, F')$ be an $(n-1)$-OMPA built from the OMPA $\mathcal{M}_{[1,n[}$ and the FSA $\mathcal{A}'$ such that $\langle (q_1, p_1), \gamma_1, \ldots, \gamma_{n-1} \rangle \overset{a}{\longrightarrow}_{\mathcal{M}'} \langle (q_2, p_2), \alpha_1, \ldots, \alpha_{n-1} \rangle$

if and only if $\langle q_1, \gamma_1, \ldots, \gamma_{n-1}, \epsilon \rangle \xrightarrow{a}_{\mathcal{M}_{[1,n[}} \langle q_2, \alpha_1, \ldots, \alpha_{n-1}, \alpha_n \rangle$ and $p_2 \stackrel{\alpha_n}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p_1$ for some $\alpha_n \in \left((\Gamma \backslash \{\bot\}) \cup \{\epsilon\} \right)$. In fact, the OMPA $\mathcal{M}'$ defines a kind of synchronous product between the pushed word over the n-th stack of OMPA $\mathcal{M}_{[1,n[}$ and the reverse of the input word of the FSA $\mathcal{A}'$. Observe that the size of the constructed $(n-1)$- OMPA $\mathcal{M}'$ is $O(|\mathcal{M}|^2)$.

Then, the relation between $\mathcal{M}'$, $\mathcal{M}_{[1,n[}$, and $\mathcal{A}'$ is given by Lemma 5.4 which follows immediately from the definition of $\mathcal{M}'$.

Lemma 5.4. $((q_1, p_1), w_1, \ldots, w_{n-1}) \stackrel{\varsigma}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}')} ((q_2, p_2), \bot, \ldots, \bot)$ iff there is a $v \in (\Gamma \backslash \{\bot\})^*$ such that $(q_1, w_1, \ldots, w_{n-1}, \bot) \stackrel{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q_2, \bot, \ldots, \bot, v\bot)$ and $p_2 \stackrel{v}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p_1$.

Now, we can apply the induction hypothesis to $\mathcal{M}'$ to show that for every $(q'', p'') \in Q \times Q_{\mathcal{A}'}$, it possible to construct, in time $O(|\mathcal{M}|^{2^{d(n-1)+2}})$, an $\mathcal{M}'$-automaton $\mathcal{A}_{(q'', p'')}$ such that $|\mathcal{A}_{(q'', p'')}| = O(|\mathcal{M}|^{2^{d(n-1)+2}})$ and $L_{\mathcal{M}'}(\mathcal{A}_{(q'', p'')}) = Pre^*_{\mathcal{T}(\mathcal{M}')}(\{((q'', p''), \bot, \ldots, \bot)\})$.

From the $\mathcal{M}'$-automata $\mathcal{A}_{(q'', p'')}$ and the $\mathcal{M}$-automaton $\mathcal{A}'$, we can construct an $\mathcal{M}$-automaton $\mathcal{A}$ such that $(q', w_1, \ldots, w_n) \in L_{\mathcal{M}}(\mathcal{A})$ if and only if there are $q'' \in Q$ and $p', p'' \in Q_{\mathcal{A}}$ such that: (1) $q'' \stackrel{\bot^{n-1}}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p''$, (2) $((q', p'), w_1, \ldots, w_{n-1}) \in L_{\mathcal{M}'}(\mathcal{A}_{(q'', p'')})$, and (3) $p' \stackrel{w_n}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p$ for some $p \in F_{\mathcal{A}'}$. Observe that such an automaton $\mathcal{A}$ of the size $O(|\mathcal{M}|^{2^{dn}})$ (by taking d as big as needed) is effectively constructible from $\mathcal{A}_{(q'', p'')}$ and $\mathcal{A}'$ using standard automata operations. Moreover, we have:

Lemma 5.5. $L_{\mathcal{M}}(\mathcal{A}) = Pre^*_{\mathcal{T}(\mathcal{M})}(\{(q, \bot, \ldots, \bot)\})$.

Proof. ($\subseteq$) Let $(q', w_1, \ldots, w_n) \in L_{\mathcal{M}}(\mathcal{A})$. Then, there are $q'' \in Q$ and $p', p'' \in Q_{\mathcal{A}}$ such that: (1) $q'' \stackrel{\bot^{n-1}}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p''$, (2) $((q', p'), w_1, \ldots, w_{n-1}) \in L_{\mathcal{M}'}(\mathcal{A}_{(q'', p'')})$, and (3) $p' \stackrel{w_n}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p$ for some $p \in F_{\mathcal{A}'}$.

So, we can apply Lemma 5.4 to the run $((q', p'), w_1, \ldots, w_{n-1}) \stackrel{\varsigma}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}')} ((q'', p''), \bot, \ldots, \bot)$ to show that there is $v \in \Gamma^*$ such that $(q', w_1, \ldots, w_{n-1}, \bot) \stackrel{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q'', \bot, \ldots, \bot, v)$ and $p'' \stackrel{v}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p'$. Thus, we have $(q', w_1, \ldots, w_{n-1}, w_n) \stackrel{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q'', \bot, \ldots, \bot, vw_n)$.

Now, we can use the runs $q'' \stackrel{\bot^{n-1}}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p''$, $p'' \stackrel{v}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p'$, and $p' \stackrel{w_n}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p$ to show that $(q'', \bot, \ldots, \bot, vw_n) \in L_{\mathcal{M}}(\mathcal{A}')$. This implies that $(q'', \bot, \ldots, \bot, vw_n) \stackrel{\tau'}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q, \bot, \ldots, \bot)$.

Hence, we have $(q', w_1, \ldots, w_n) \in Pre^*_{\mathcal{T}(\mathcal{M})}(\{(q, \bot, \ldots, \bot)\})$ and therefore $L_{\mathcal{M}}(\mathcal{A}) \subseteq Pre^*_{\mathcal{T}(\mathcal{M})}(\{(q, \bot, \ldots, \bot)\})$.

($\supseteq$) Let $(q', w_1, \ldots, w_n) \in Pre^*_{\mathcal{T}(\mathcal{M})}(\{(q, \bot, \ldots, \bot)\})$. Then, there are $q'' \in Q$, $v \in \Gamma^*$, and $\tau, \tau' \in \Sigma^*$ such that:

$$(q', w_1, \ldots, w_n) \stackrel{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q'', \bot, \ldots, \bot, vw_n) \stackrel{\tau'}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q, \bot, \ldots, \bot)$$

Since $(q'', \bot, \ldots, \bot, vw_n) \stackrel{\tau'}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M})} (q, \bot, \ldots, \bot)$, we have $(q'', \bot, \ldots, \bot, vw_n) \in L_{\mathcal{M}}(\mathcal{A}')$. This implies that there are $p', p'' \in Q_{\mathcal{A}'}$ and $p \in F_{\mathcal{A}'}$ such that $q'' \stackrel{\bot^{n-1}}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p''$, $p'' \stackrel{v}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p'$, and $p' \stackrel{w_n}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p$.

On the other hand, we can show $(q', w_1, \ldots, w_{n-1}, \perp) \overset{\tau}{\Longrightarrow}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q'', \perp, \ldots, \perp, v)$ since we have $(q', w_1, \ldots, w_n) \overset{\tau}{\Longrightarrow}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q'', \perp, \ldots, \perp, vw_n)$.

Then, we can apply Lemma 5.4 to $(q', w_1, \ldots, w_{n-1}, \perp) \overset{\tau}{\Longrightarrow}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q'', \perp, \ldots, \perp, v)$ and $p'' \overset{v}{\Longrightarrow}^*_{\mathcal{T}(\mathcal{A'})} p'$ to show that $((q', p'), w_1, \ldots, w_{n-1}) \overset{\varsigma}{\Longrightarrow}^*_{\mathcal{T}(\mathcal{M'})} ((q'', p''), \perp, \ldots, \perp)$. This implies that $((q', p'), w_1, \ldots, w_{n-1}) \in L_{\mathcal{M'}}(\mathcal{A}_{(q'', p'')})$. Now, we can use the definition of the $\mathcal{M}$-automaton $\mathcal{A}$ to show that $(q', w_1, \ldots, w_n) \in L_{\mathcal{M}}(\mathcal{A})$ since we have $q'' \overset{\perp^{n-1}}{\Longrightarrow}^*_{\mathcal{T}(\mathcal{A'})} p''$, $((q', p'), w_1, \ldots, w_{n-1}) \in L_{\mathcal{M'}}(\mathcal{A}_{(q'', p'')})$, and $p' \overset{w_n}{\Longrightarrow}^*_{\mathcal{T}(\mathcal{A'})} p$ with $p \in F_{\mathcal{A'}}$. Hence, we have $L_{\mathcal{M}}(\mathcal{A}) \supseteq Pre^*_{\mathcal{T}(\mathcal{M})}(\{(q, \perp, \ldots, \perp)\})$.   □

This terminates the proof of Theorem 5.3.   □

As an immediate consequence of Theorem 5.3 and Lemma 5.1, we obtain:

Theorem 5.6. *Let $\mathcal{M} = (n, Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0, F)$ be an OMPA and $\mathcal{A}'$ be an $\mathcal{M}$-automaton. Then, it is possible to construct, in time $O((|\mathcal{M}| \cdot |\mathcal{A}'|)^{2^{dn}})$ where d is a constant, an $\mathcal{M}$-automaton $\mathcal{A}$ such that $|\mathcal{A}| = O((|\mathcal{M}| \cdot |\mathcal{A}'|)^{2^{dn}})$ and $L_{\mathcal{M}}(\mathcal{A}) = Pre^*_{\mathcal{T}(\mathcal{M})}(L_{\mathcal{M}}(\mathcal{A}'))$.*

We can extend the previous result to show that the operator Pre^+ preserves also recognizability.

Theorem 5.7. *Let $\mathcal{M} = (n, Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0, F)$ be an OMPA and $\mathcal{A}'$ be an $\mathcal{M}$-automaton. Then, it is possible to construct, in time $O((|\mathcal{M}| \cdot |\mathcal{A}'|)^{2^{dn}})$ where d is a constant, an $\mathcal{M}$-automaton $\mathcal{A}$ such that $|\mathcal{A}| = O((|\mathcal{M}| \cdot |\mathcal{A}'|)^{2^{dn}})$ and $L_{\mathcal{M}}(\mathcal{A}) = Pre^+_{\mathcal{T}(\mathcal{M})}(L_{\mathcal{M}}(\mathcal{A}'))$.*

Proof. In the following, we show that computing the set $Pre^+_{\mathcal{T}(\mathcal{M})}(L_{\mathcal{M}}(\mathcal{A}'))$ can be reduced to computing the set $Pre^*_{\mathcal{T}(\mathcal{M'})}(L_{\mathcal{M'}}(\mathcal{A}''))$ for an OMPA $\mathcal{M}'$ and an $\mathcal{M}$-automaton $\mathcal{A}''$ such that $|\mathcal{M}'| = O(|\mathcal{M}|)$ and $|\mathcal{A}''| = O(|\mathcal{A}|)$. Intuitively, the OMPA has the same stack and input alphabets as the ones of $\mathcal{M}$. Corresponding to each state q of $\mathcal{M}$, $\mathcal{M}'$ has q and q_{copy} as two states where q_{copy} is a fresh symbol which was not used neither in the definition of $\mathcal{M}$ nor in the definition of $\mathcal{A}$. For any transition of the form $\langle q, \gamma_1, \ldots, \gamma_n \rangle \overset{a}{\longrightarrow}_{\mathcal{M}} \langle q', \alpha_1, \ldots, \alpha_n \rangle$, $\mathcal{M}'$ has two transitions $\langle q, \gamma_1, \ldots, \gamma_n \rangle \overset{a}{\longrightarrow}_{\mathcal{M'}} \langle q', \alpha_1, \ldots, \alpha_n \rangle$ and $\langle q, \gamma_1, \ldots, \gamma_n \rangle \overset{a}{\longrightarrow}_{\mathcal{M'}} \langle q'_{copy}, \alpha_1, \ldots, \alpha_n \rangle$. Any computation of $\mathcal{M}'$ can be divided in two phases. In the first phase $\mathcal{M}'$ mimics the behavior of the OMPA $\mathcal{M}'$ by performing the same sequence of transitions. In the second phase, the OMPA $\mathcal{M}'$ performs a transition from a state $q \in Q$ of $\mathcal{M}$ to a state q'_{copy} with $q' \in Q$ and halts. Formally, $\mathcal{M}'$ is defined by the tuple $(n, Q \cup Q_{copy}, \Sigma, \Gamma, \Delta \cup \Delta', q_0, \gamma_0, F)$ where $Q_{copy} = \{q_{copy} \mid q \in Q\}$ and $\Delta' = \{\langle q, \gamma_1, \ldots, \gamma_n \rangle \overset{a}{\longrightarrow}_{\mathcal{M'}} \langle q'_{copy}, \alpha_1, \ldots, \alpha_n \rangle \mid \langle q, \gamma_1, \ldots, \gamma_n \rangle \overset{a}{\longrightarrow}_{\mathcal{M}} \langle q', \alpha_1, \ldots, \alpha_n \rangle\}$.

Let $\mathcal{A}' = (Q_{\mathcal{M}}, \Gamma, \Delta_{\mathcal{M}}, I_{\mathcal{M}}, F_{\mathcal{M}})$ be the $\mathcal{M}$-automaton. We assume here that $\mathcal{A}$ has no transition leading to an initial state. Now, we can construct the $\mathcal{M}'$-automaton $\mathcal{A}'' = (Q_{\mathcal{M'}}, \Gamma, \Delta_{\mathcal{M'}}, I_{\mathcal{M'}}, F_{\mathcal{M'}})$ from the $\mathcal{M}$-automaton $\mathcal{A}'$ as follows: The set of states of $\mathcal{A}''$ is the union of the set of states of $\mathcal{A}'$ and the set of states of $\mathcal{M}'$ (i.e., $Q_{\mathcal{M'}} = Q_{\mathcal{M}} \cup Q_{copy}$). The set of transitions of $\mathcal{A}''$ contains any transition of $\mathcal{A}$ that does not involve a state of Q (i.e., $(\Delta_{\mathcal{M}} \setminus (Q \times \Gamma_\epsilon \times Q_{\mathcal{M}})) \subseteq \Delta_{\mathcal{M'}}$). Moreover, corresponding to any transition of the form $q \overset{a}{\longrightarrow}_{\mathcal{A'}} p$ where $q \in Q$, the automaton $\mathcal{A}''$ has a transition of the form $q_{copy} \overset{a}{\longrightarrow}_{\mathcal{A''}} p$. That is, the automaton $\mathcal{A}''$ is precisely $\mathcal{A}'$ where any initial state $q \in Q$ is relabeled by its

copy q_{copy}. (Observe that there is no transition from/to a state $q \in Q$ in $\mathcal{A}''$.) Since $\mathcal{A}''$ is an $\mathcal{M}'$-automaton, we have $I_{\mathcal{M}'} = Q \cup Q_{copy}$. We have also $F_{\mathcal{M}'} = F_{\mathcal{M}}$.

Then it is easy to see that the set $Pre^+_{\mathcal{T}(\mathcal{M})}(L_{\mathcal{M}}(\mathcal{A}'))$ is precisely the set $Pre^*_{\mathcal{T}(\mathcal{M}')}(L_{\mathcal{M}'}(\mathcal{A}'')) \cap Conf(\mathcal{M})$. Thus, we can apply Theorem 5.6 to show that it is possible to construct, in time $O((|\mathcal{M}| \cdot |\mathcal{A}'|)^{2^{dn}})$ where d is a constant, an $\mathcal{M}$-automaton $\mathcal{A}$ such that $|\mathcal{A}| = O((|\mathcal{M}| \cdot |\mathcal{A}'|)^{2^{dn}})$ and $L_{\mathcal{M}}(\mathcal{A}) = Pre^+_{\mathcal{T}(\mathcal{M})}(L_{\mathcal{M}}(\mathcal{A}'))$. $\qquad\square$

6. Linear-Time Global Model Checking

In this section, we show that the model-checking problem of ω-regular properties for OMPA is decidable and in 2ETIME. Observe that this result subsumes the 2ETIME upper bound obtained for the emptiness problem of OMPA (see Theorem 4.6). In fact, we can see the emptiness problem (i.e., the reachability problem) of OMPA as a particular instance of the LTL-model checking problem of OMPA for which the decision procedure (provided in Section 4) is simpler.

To prove the 2ETIME upper bound for the model-checking problem of ω-regular properties for OMPA, we introduce the repeated state reachability problem for OMPA.

We fix an OMPA $\mathcal{M} = (n, Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0, F)$ for the rest of the paper such that $\Sigma = \Delta$ and $t = ((q, \gamma_1, \ldots, \gamma_n), a, (q', \alpha_1, \ldots, \alpha_n))$ is in Δ if and only if $a = t$.

6.1. **The repeated state reachability problem.** In the following, we are interested in solving *the repeated state reachability problem* which consists in computing, for a given state $q_f \in Q$, the set of all configurations c of $\mathcal{M}$ such that there is an infinite run of $\mathcal{T}(\mathcal{M})$ starting from c that visits infinitely often the state q_f.

To this aim, let us introduce the following notation: For every $i \in [1, n]$, we denote by $\mathcal{M}_{[1,i]} = (n, Q, \Sigma, \Gamma, \Delta_{[1,i]}, q_0, \gamma_0, F)$ the OMPA built from $\mathcal{M}$ by discarding pop transitions of $\mathcal{M}$ over the last $(n - i)$ stacks. Formally, we have $\Delta_{[1,i]} = \Delta \cap ((Q \times (\Gamma_\epsilon)^i \times (\{\epsilon\})^{n-i}) \times \Sigma \times (Q \times (\Gamma^*)^n))$.

For every $i \in [1, n]$, and every $(q, \gamma) \in Q \times (\Gamma \setminus \{\bot\})$, let $C_i^{(q,\gamma)}$ denote the set of all configurations $(q, w_1, \ldots, w_n) \in Conf(\mathcal{M})$ such that $w_1 = \cdots = w_{i-1} = \bot$ and $w_i = \gamma u$ for some $u \in Stack(\mathcal{M})$. Moreover, let $c_i^{(q,\gamma)}$ be the configuration $(q, w_1, \ldots, w_n)$ of $\mathcal{M}$ such that $w_i = \gamma \bot$ and $w_j = \bot$ for all $j \neq i$.

In the following, we show that detecting an infinite computation of $\mathcal{M}$ that visits infinitely often a state q_f can be reduced to detecting an infinite computation the form $\rho_1 \cdot \rho_2^\omega$ that eventually repeats the same sequence of transitions indefinitely and visits q_f infinitely often (and where ρ_1 and ρ_2 are finite computations). Hence, a periodic computation is a run which, after a finite computation prefix ρ_1 (called stem), ultimately repeats the same sequence of transitions ρ_2 (called lasso) over and over. Let us give some intuitions behind this reduction.

Let us assume that there is an infinite computation ρ of $\mathcal{T}(\mathcal{M})$ starting from a configuration c of $\mathcal{M}$. Let i be the maximal index of the stack that is popped infinitely often. This means that, at some point of the computation, the stacks from $(n - i + 1)$ to n will never be popped. Let us concentrate on the suffix of the computation ρ which contains only push transitions on the stacks from $(n-i+1)$ to n. Let $c_1 c_2 \cdots$ be the sequence of configurations in this suffix of ρ where the first $(i - 1)$ stacks are empty. Applying a similar argument

to the content of the ith-stack, along the sequence of configurations $c_1 c_2 \cdots$, as the one for standard pushdown automata [BEM97], we can deduce that the ith-stack is *increasing*. This means that there are indices $j_1 < j_2$, a stack symbol $\gamma \in \Gamma$, and a state $q \in Q$ such that the configurations c_{j_1} and c_{j_2} are in $C_i^{(q,\gamma)}$ and the symbol γ at the top of the ith-stack in c_{j_1} and c_{j_2} will never be popped. Furthermore, along the sub-computation ρ_2 from c_{j_1} and c_{j_2} the state q_f is visited. Observe that if we remove from the configuration c_{j_1} all the stack symbols that will never be popped we obtain the configuration $c_i^{(q,\gamma)}$. Then, the computation ρ_2 can be simplified, by dropping all the (useless) stack symbols that will never be popped from the configuration c_{j_1}, as follows : $c_i^{(q,\gamma)} \overset{\tau_1,+}{\Longrightarrow}_{\mathcal{T}(\mathcal{M}_{[1,i]})}(q_f, w_1, \ldots, w_2) \overset{\tau_2,*}{\Longrightarrow}_{\mathcal{T}(\mathcal{M}_{[1,i]})} c_2'$ with $c_2' \in C_i^{(q,\gamma)}$. This computation ρ_2 represents our lasso computation since from the configuration c_2' we can repeat the sequence of transitions $\tau_1 \tau_2$ while visiting the same states (and in particular the state q_f).

The existence of a such lasso is expressed by the second item of Theorem 6.1 while the existence of a stem computation from the starting configuration c to the configuration $c_{j_1} \in C_i^{(q,\gamma)}$ is stated by the first item of Theorem 6.1.

Then, the solution of the *repeated state reachability problem* is formally based on the following fact:

Theorem 6.1. *Let c be a configuration of $\mathcal{M}$ and q_f be a state of $\mathcal{M}$. There is an infinite run starting from c that visits infinitely often the state q_f if and only if there are $i \in [1, n]$, $q \in Q$, and $\gamma \in \Gamma$ such that:*

(1) $c \in Pre_{\mathcal{T}(\mathcal{M})}^*(C_i^{(q,\gamma)})$, and

(2) $c_i^{(q,\gamma)} \in Pre_{\mathcal{T}(\mathcal{M}_{[1,i]})}^+\big(Pre_{\mathcal{T}(\mathcal{M}_{[1,i]})}^*(C_i^{(q,\gamma)}) \cap (\{q_f\} \times (Stack(\mathcal{M}))^n)\big).$

Proof. $(\Rightarrow)$: Let $\rho = c_0 t_0 c_1 t_1 c_2 t_2 \cdots$ be an infinite computation of $\mathcal{T}(\mathcal{M})$ starting from the configuration $c_0 = c$ of $\mathcal{M}$. For every $j \in \mathbb{N}$, c_j is a configuration of $\mathcal{M}$ and t_j is a transition of $\mathcal{M}$ such that $c_j \overset{t_j}{\longrightarrow}_{\mathcal{T}(\mathcal{M})} c_{j+1}$ (recall that $\Sigma = \Delta$). Let $i \in [1, n]$ be the maximal index such that for every $j \in \mathbb{N}$, there is $k_j \geq j$ such that t_{k_j} is a pop transition over the ith stack of $\mathcal{M}$. This implies that c_{k_j} is in $Q \times (\{\bot\})^{i-1} \times ((\Gamma \setminus \{\bot\})^* \cdot Stack(\mathcal{M})) \times (Stack(\mathcal{M}))^{n-i}$ (i.e., the first $(i-1)$-stacks are empty) since t_{k_j} is a pop transition from the ith stack of $\mathcal{M}$.

From the definition of i, there is $r \in \mathbb{N}$ such that for every $h \geq r$, there is $d_h \in [1, i]$ such that the transition t_h is a pop transition over the stack d_h of $\mathcal{M}$ (i.e., the transition t_h is not a pop transition from the stack from $(n - i + 1)$ to n). This implies that for every $h \geq r$, we have $c_h \overset{t_h}{\longrightarrow}_{\mathcal{T}(\mathcal{M}_{[1,i]})} c_{h+1}$.

Then, we construct a sequence $\pi = c_{j_0} c_{j_1} c_{j_2} \cdots$ of configurations of $\mathcal{M}$ as follows: c_{j_0} is the first configuration of ρ such that $j_0 \geq r$ and t_{j_0} is a pop transition over the ith-stack of $\mathcal{M}$, for every $\ell > 0$, c_{j_ℓ} is the first configuration of ρ such that $j_\ell > j_{\ell-1}$ and t_{j_ℓ} is a pop transition over the i-stack of $\mathcal{M}$. Recall that, by definition, we have for every $l \in \mathbb{N}$, c_{j_l} is in $Q \times (\{\bot\})^{i-1} \times ((\Gamma \setminus \{\bot\})^* \cdot Stack(\mathcal{M})) \times (Stack(\mathcal{M}))^{n-i}$ (i.e., the first $(i-1)$ stacks are empty).

Now, for every $l \geq 0$, let $\pi^{(l)}$ be the suffix of π starting at c_{j_l}, and let $m^{(l)}$ be the minimal length of the configurations of $\pi^{(l)}$, where the length of a configuration is defined as the length of its ith stack.

Construct a subsequence $\pi' = c_{z_0} c_{z_1} c_{z_2} \cdots$ of π as follows: c_{z_0} is the first configuration of π of length $m^{(0)}$; for every $l > 0$, c_{z_l} is the first configuration of $\pi^{(z_{l-1}+1)}$ of length $m^{(z_{l-1}+1)}$.

Since the number of states and stack symbols is finite, there exists a subsequence $\pi'' = c_{x_0} c_{x_1} c_{x_2} \cdots$ of π' whose elements have all the same state q, and the same symbol γ on the top of the ith stack. Observe that $c_{x_0}, c_{x_1}, c_{x_2}, \ldots$ are in $C_i^{(q,\gamma)}$.

Since ρ is an accepting run, there is an index $b \geq 1$ and a configuration c_{q_f} with state q_f such that:

$$c_0 \overset{\tau}{\underset{\mathcal{T}(\mathcal{M})}{\Longrightarrow}}{}^* c_{x_0} \overset{\tau'}{\underset{\mathcal{T}(\mathcal{M})}{\Longrightarrow}}{}^+ c_{q_f} \overset{\tau''}{\underset{\mathcal{T}(\mathcal{M})}{\Longrightarrow}}{}^* c_{x_b}$$

Since $c_0 = c$ and $c_{x_0} \in C_i^{(q,\gamma)}$, we have $c \in Pre^*_{\mathcal{T}(\mathcal{M})}(C_i^{(q,\gamma)})$, and so (1) holds.

Due to the definition of π (and so, π' and π''), we have

$$c_{x_0} \overset{\tau'}{\underset{\mathcal{T}(\mathcal{M}_{[1,i]})}{\Longrightarrow}}{}^+ c_{q_f} \overset{\tau''}{\underset{\mathcal{T}(\mathcal{M}_{[1,i]})}{\Longrightarrow}}{}^* c_{x_b}$$

Since $c_{x_0} \in Q \times (\{\bot\})^{i-1} \times ((\Gamma \setminus \{\bot\})^* \cdot Stack(\mathcal{M})) \times (Stack(\mathcal{M}))^{n-i}$, then there are $w_i, w_{i+1}, \ldots, w_n \in Stack(\mathcal{M})$ such that $c_{x_0} = (q, \bot, \ldots, \bot, \gamma w_i, w_{i+1}, \ldots, w_n)$. Due to the definition of the subsequence π' and π'' all the configurations of ρ between c_{x_0} and c_{x_b} have a content of the l-th stack (with $i \leq l \leq k$) of the form $w'_l w_l$. In particular, the configuration c_{q_f} is of the form $(q_f, u_1, \ldots, u_{i-1}, u_i w_i, u_{i+1} w_{i+1}, \ldots, u_n w_n)$ and the configuration c_{x_b} is of the form $(q, \bot, \ldots, \bot, \gamma v_i w_i, v_{i+1} w_{i+1}, \ldots, v_n w_n)$. This implies:

$$c_i^{(q,\gamma)} = (q, \bot, \ldots, \bot, \gamma, \bot, \ldots, \bot) \overset{\tau'}{\underset{\mathcal{T}(\mathcal{M}_{[1,i]})}{\Longrightarrow}}{}^+ (q_f, u_1, \ldots, u_{i-1}, u_i, u_{i+1}, \ldots, u_n)$$

and

$$(q_f, u_1, \ldots, u_{i-1}, u_i, u_{i+1}, \ldots, u_n) \overset{\tau''}{\underset{\mathcal{T}(\mathcal{M}_{[1,i]})}{\Longrightarrow}}{}^* (q, \bot, \ldots, \bot, \gamma v_i, v_{i+1}, \ldots, v_n)$$

Consequently, (2) holds, which concludes the proof.

($\Leftarrow$) : We can use (1) and (2) of Theorem 6.1 to construct a run starting from c that visits infinitely often the state q_f. $\qquad\square$

Since the sets of configurations $C_i^{(q,\gamma)}$ and $(\{q_f\} \times (Stack(\mathcal{M}))^n)$ are recognizable, we can use Theorem 5.6 and Theorem 5.7 to construct $\mathcal{M}$-automata recognizing $Pre^*_{\mathcal{T}(\mathcal{M})}(C_i^{(q,\gamma)})$ and $Pre^+_{\mathcal{T}(\mathcal{M}_{[1,i]})}\big(Pre^*_{\mathcal{T}(\mathcal{M}_{[1,i]})}(C_i^{(q,\gamma)}) \cap (\{q\} \times (Stack(\mathcal{M}))^n)\big)$. Hence, we can construct a $\mathcal{M}$-automaton that recognizes the set of all configurations c of $\mathcal{M}$ such that there is an infinite run of $\mathcal{T}(\mathcal{M})$ starting from c that visits infinitely often the state q_f.

Theorem 6.2. *Let $\mathcal{M} = (n, Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0, F)$ be an OMPA and $q_f \in Q$ be a state. Then, it is possible to construct, in time $O((|\mathcal{M}|)^{2^{dn}})$ where d is a constant, an $\mathcal{M}$-automaton $\mathcal{A}$ such that $|\mathcal{A}| = O((|\mathcal{M}|)^{2^{dn}})$ and for every configuration $c \in Conf(\mathcal{M})$, $c \in L_{\mathcal{M}}(\mathcal{A})$ if and only if there is an infinite run of $\mathcal{T}(\mathcal{M})$ starting from c that visits q_f infinitely often.*

Proof. We know from Theorem 6.1 that there is an infinite run of $\mathcal{T}(\mathcal{M})$ starting from c that visits q_f infinitely often if and only if $c \in Pre^*_{\mathcal{T}(\mathcal{M})}(C_i^{(q,\gamma)})$, and $c_i^{(q,\gamma)} \in Pre^+_{\mathcal{T}(\mathcal{M}_{[1,i]})}\big(Pre^*_{\mathcal{T}(\mathcal{M}_{[1,i]})}(C_i^{(q,\gamma)}) \cap (\{q_f\} \times (Stack(\mathcal{M}))^n)\big)$. (Observe that it is possible to

construct an $\mathcal{M}$-automaton representing the set $C_i^{(q,\gamma)}$ and which size is linear in the size of $\mathcal{M}$.)

Then, for every index $i \in [1,n]$, state $q \in Q$ and stack symbol $\gamma \in \Gamma$, we construct an $\mathcal{M}$-automaton $\mathcal{A}_1^{(q,i,\gamma)}$ recognizing the set $Pre_{\mathcal{T}(\mathcal{M}_{[1,i]})}^*(C_i^{(q,\gamma)})$ and such that $|\mathcal{A}_1^{(q,i,\gamma)}| = O((|\mathcal{M}|)^{2^{d'n}})$ where d' is a constant. From Theorem 5.6, we know that such an automaton $\mathcal{A}_1^{(q,i,\gamma)}$ can be constructed in time $O((|\mathcal{M}|)^{2^{d'n}})$.

Now, we can construct an $\mathcal{M}$-automaton $\mathcal{A}_2^{(q,i,\gamma)}$ recognizing precisely the set $\left(Pre_{\mathcal{T}(\mathcal{M}_{[1,i]})}^*(C_i^{(q,\gamma)}) \cap (\{q_f\} \times (Stack(\mathcal{M}))^n)\right)$ and such that $|\mathcal{A}_2^{(q,i,\gamma)}| = O((|\mathcal{M}|)^{2^{d'n}})$. Observe that $\mathcal{A}_2^{(q,i,\gamma)}$ can be constructed in time $O((|\mathcal{M}|)^{2^{d'n}})$ from the $\mathcal{M}$-automaton $\mathcal{A}_1^{(q,i,\gamma)}$.

We can apply Theorem 5.7 to $\mathcal{M}$ and $\mathcal{A}_2^{(q,i,\gamma)}$ to show that we can construct an $\mathcal{M}$-automaton $\mathcal{A}_3^{(q,i,\gamma)}$ recognizing $Pre_{\mathcal{T}(\mathcal{M}_{[1,i]})}^+\left(Pre_{\mathcal{T}(\mathcal{M}_{[1,i]})}^*(C_i^{(q,\gamma)}) \cap (\{q_f\} \times (Stack(\mathcal{M}))^n)\right)$ and such that $|\mathcal{A}_3^{(q,i,\gamma)}| = O((|\mathcal{M}|)^{2^{d''n}})$ for some constant $d'' > d'$. Moreover, such an $\mathcal{M}$-automaton $\mathcal{A}_3^{(q,i,\gamma)}$ can be constructed in time $O((|\mathcal{M}|)^{2^{d''n}})$. Then, checking whether $c_i^{(q,\gamma)}$ is in $L_{\mathcal{M}}(\mathcal{A}_3^{(q,i,\gamma)})$ can be performed in time polynomial in $|\mathcal{A}_3^{(q,i,\gamma)}|$ [HU79].

If $c_i^{(q,\gamma)}$ is not in $L_{\mathcal{M}}(\mathcal{A}_3^{(q,i,\gamma)})$ then let $\mathcal{A}^{(q,i,\gamma)}$ be the $\mathcal{M}$-automaton recognizing the empty set (i.e., $L_{\mathcal{M}}(\mathcal{A}^{(q,i,\gamma)}) = \emptyset$). Otherwise let $\mathcal{A}^{(q,i,\gamma)}$ be the $\mathcal{M}$-automaton recognizing the set $Pre_{\mathcal{T}(\mathcal{M})}^*(C_i^{(q,\gamma)})$ and such that $|\mathcal{A}^{(q,i,\gamma)}| = O((|\mathcal{M}|)^{2^{d'n}})$. From Theorem 5.6, we know that such an automaton $\mathcal{A}^{(q,i,\gamma)}$ can be constructed in time $O((|\mathcal{M}|)^{2^{d'n}})$.

By taking d as big as needed, we can construct, in time $O((|\mathcal{M}|)^{2^{dn}})$ where d is a constant, the $\mathcal{M}$-automaton $\mathcal{A}$ such that $|\mathcal{A}| = O((|\mathcal{M}|)^{2^{dn}})$ and for every configuration $c \in Conf(\mathcal{M})$, $c \in L_{\mathcal{M}}(\mathcal{A})$ if and only if there is an infinite run of $\mathcal{T}(\mathcal{M})$ starting from c that visits q_f infinitely often. The $\mathcal{M}$-automaton $\mathcal{M}$ is just the union of all the $\mathcal{M}$-automata $\mathcal{A}^{(q,i,\gamma)}$. $\qquad\square$

6.2. w-regular properties.

In the following, we assume that the reader is familiar with w-regular properties expressed in the linear-time temporal logics [Pnu77] or the linear time μ-calculus [Var88]. For more details, the reader is referred to [Pnu77, VW86, Var88, Var95].

Let φ be an w-regular formula built from a set of atomic propositions $Prop$, and let $\mathcal{M} = (n, Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0, F)$ be an OMPA with a labeling function $\Lambda : Q \to 2^{Prop}$ associating to each state $q \in Q$ the set of atomic propositions that are true in it. Afterwards, we are interested in solving *the global model checking problem* which consists in computing the set of all configurations c of $\mathcal{M}$ such that every infinite run starting from c satisfies φ.

To solve this problem, we adopt an approach similar to [BM96, BEM97] and we construct a Buchi automaton $\mathcal{B}_{\neg\varphi}$ over the alphabet 2^{Prop} accepting the negation of φ [VW86, Var95]. Then, we compute the product of the OMPA $\mathcal{M}$ and of the Büchi automaton $\mathcal{B}_{\neg\varphi}$ to obtain an n-OMPA $\mathcal{M}_{\neg\varphi}$ with a set of repeating states G. Now, it is easy to see that the original problem can be reduced to the *repeated state reachability problem* which compute the set of all configurations c such that there is an infinite run of $\mathcal{T}(\mathcal{M})$ starting from c that visits infinitely often a state in G. Hence, as an immediate consequence of Theorem 6.2, we obtain:

Theorem 6.3. *Let $\mathcal{M} = (n, Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0, F)$ be an OMPA with a labeling function Λ, and let φ be a linear time μ-calculus formula or linear time temporal formula. Then, it is possible to construct, in time $O((2^{|\varphi|} \cdot |\mathcal{M}|)^{2^{dn}})$ where d is a constant, an $\mathcal{M}$-automaton $\mathcal{A}$ such that $|\mathcal{A}| = O((2^{|\varphi|} \cdot |\mathcal{M}|)^{2^{dn}})$ and for every configuration $c \in Conf(\mathcal{M})$, $c \in L_{\mathcal{M}}(\mathcal{A})$ if and only if there is an infinite run of $\mathcal{T}(\mathcal{M})$ starting from c does not satisfy φ.*

Proof. It is well known that it is possible to construct, in time exponential in $|\varphi|$, a Büchi automaton $\mathcal{B}_{\neg\varphi}$ for the negation of φ having exponential size in $|\varphi|$ [VW86, Var88]. Therefore, the product of $\mathcal{M}$ and $\mathcal{B}_{\neg\varphi}$ has polynomial size in $|\mathcal{M}|$ and exponential size in $|\varphi|$. Applying Theorem 6.2 to the n-OMPA $\mathcal{M}_{\neg\varphi}$ (the product of $\mathcal{M}$ and $\mathcal{B}_{\neg\varphi}$) of size $O(2^{|\varphi|} \cdot |\mathcal{M}|)$ we obtain our complexity result. ◻

Observe that we can also construct an $\mathcal{M}$-automaton $\mathcal{A}'$ such that for every configuration $c \in Conf(\mathcal{M})$, $c \in L_{\mathcal{M}}(\mathcal{A})$ if and only if every infinite run of $\mathcal{T}(\mathcal{M})$ starting from c that satisfies φ since the class of $\mathcal{M}$-automata is closed under boolean operations.

We are now ready to establish our result about the model checking problem for w-regular properties which consists in checking whether, for a given configuration c of the OMPS, every infinite run starting from c satisfies the formula φ.

Theorem 6.4. *The model checking problem for the linear-time temporal logics or the linear-time μ-calculus and OMPA is 2ETIME-complete.*

Proof. The 2ETIME upper bound is established by Theorem 6.3. To prove hardness, we use the fact that the emptiness problem for ordered multi-pushdown automata is 2ETIME-complete [ABH08]. ◻

7. Conclusion

We have shown that the set of all predecessors of a recognizable set of configurations of an ordered multi-pushdown automaton is an effectively constructible recognizable set. We have also proved that the set of all configurations of an ordered multi-pushdown automaton that satisfy a given w-regular property is effectively recognizable. From these results we have derived an 2ETIME upper bound for the model checking problem of w-regular properties.

It may be interesting to see if our approach can be extended to solve the global model-checking problem for branching time properties expressed in CTL or CTL* by adapting the constructions given in [BEM97, FWW97] for standard pushdown automata.

References

[ABH08] M. F. Atig, B. Bollig, and P. Habermehl. Emptiness of multi-pushdown automata is 2ETIME-complete. In *Proceedings of DLT'08*, volume 5257 of *LNCS*, pages 121–133. Springer, 2008.

[ABQ09] M. Faouzi Atig, A. Bouajjani, and S. Qadeer. Context-bounded analysis for concurrent programs with dynamic creation of threads. In *TACAS*, volume 5505 of *LNCS*, pages 107–123. Springer, 2009.

[ABT08] M. F. Atig, A. Bouajjani, and T. Touili. On the reachability analysis of acyclic networks of pushdown systems. In *CONCUR*, volume 5201 of *LNCS*, pages 356–371. Springer, 2008.

[AT09] M. F. Atig and T. Touili. Verifying parallel programs with dynamic communication structures. In *CIAA*, volume 5642 of *LNCS*, pages 145–154. Springer, 2009.

[Ati10a] M. F. Atig. From multi to single stack automata. In *Proceedings of the 21th International Conference on Concurrency Theory (CONCUR'10)*, LNCS. Springer, 2010.

[Ati10b]   Mohamed Faouzi Atig. Global model checking of ordered multi-pushdown systems. In *FSTTCS*, volume 8 of *LIPIcs*, pages 216–227. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.

[BCCC96]   L. Breveglieri, A. Cherubini, C. Citrini, and S. Crespi Reghizzi. Multi-push-down languages and grammars. *International Journal of Foundations of Computer Science*, 7(3):253–292, 1996.

[BEM97]   A. Bouajjani, J. Esparza, and O. Maler. Reachability analysis of pushdown automata: Application to model-checking. In *CONCUR*, volume 1243 of *LNCS*, pages 135–150. Springer, 1997.

[BEP11]   Ahmed Bouajjani, Michael Emmi, and Gennaro Parlato. On sequentializing concurrent programs. In *SAS '11: Proc. 18th Intl. Symp. on Static Analysis*, pages 129–145. Springer, 2011.

[BESS05]   Ahmed Bouajjani, Javier Esparza, Stefan Schwoon, and Jan Strejcek. Reachability analysis of multithreaded software with asynchronous communication. In *FSTTCS*, volume 3821 of *LNCS*, pages 348–359. Springer, 2005.

[BFQ07]   Ahmed Bouajjani, Séverine Fratani, and Shaz Qadeer. Context-bounded analysis of multi-threaded programs with dynamic linked structures. In *CAV*, volume 4590 of *Lecture Notes in Computer Science*, pages 207–220, 2007.

[BM96]   A. Bouajjani and O. Maler. Reachability analysis of pushdown automata. In *Proc. Intern. Workshop on Verification of Infinite-State Systems (Infinity'96)*, 1996.

[BMOT05]   A. Bouajjani, M. Müller-Olm, and T. Touili. Regular symbolic analysis of dynamic networks of pushdown systems. In *CONCUR'05*, LNCS, 2005.

[BT03]   A. Bouajjani and T. Touili. Reachability Analysis of Process Rewrite Systems. In *FSTTCS'03*. LNCS 2914, 2003.

[EHRS00]   J. Esparza, D. Hansel, P. Rossmanith, and S. Schwoon. Efficient algorithms for model checking pushdown systems. In *Proc. of CAV'2000*, number 1855 in LNCS, pages 232–247. Springer-Verlag, 2000.

[EK99]   J. Esparza and J. Knoop. An automata-theoretic approach to interprocedural data-flow analysis. In *FoSSaCS*, volume 1578 of *LNCS*, pages 14–30. Springer, 1999.

[EP00]   J. Esparza and A. Podelski. Efficient algorithms for pre* and post* on interprocedural parallel flow graphs. In *POPL'00*. ACM, 2000.

[EQR11]   Michael Emmi, Shaz Qadeer, and Zvonimir Rakamarić. Delay-bounded scheduling. In *POPL '11: Proc. 38th ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages*, pages 411–422. ACM, 2011.

[FWW97]   Alain Finkel, Bernard Willems, and Pierre Wolper. A direct symbolic approach to model checking pushdown systems (extended abstract). In Faron Moller, editor, *Proceedings of the 2nd International Workshop on Verification of Infinite State Systems (INFINITY'97)*, volume 9 of *Electronic Notes in Theoretical Computer Science*, pages 27–39, Bologna, Italy, July 1997. Elsevier Science Publishers.

[GMM10]   Pierre Ganty, Rupak Majumdar, and Benjamin Monmege. Bounded underapproximations. In *CAV*, volume 6174 of *Lecture Notes in Computer Science*, pages 600–614. Springer, 2010.

[HLMS10]   Alexander Heußner, Jérôme Leroux, Anca Muscholl, and Grégoire Sutre. Reachability analysis of communicating pushdown systems. In *FOSSACS*, volume 6014 of *Lecture Notes in Computer Science*, pages 267–281. Springer, 2010.

[HU79]   John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

[JM07]   Ranjit Jhala and Rupak Majumdar. Interprocedural analysis of asynchronous programs. In *POPL*. IEEE, 2007.

[Kah09]   V. Kahlon. Boundedness vs. unboundedness of lock chains: Characterizing decidability of pairwise cfl-reachability for threads communicating via locks. In *LICS*, pages 27–36. IEEE Computer Society, 2009.

[LMP09]   Salvatore La Torre, Parthasarathy Madhusudan, and Gennaro Parlato. Analyzing recursive programs using a fixed-point calculus. In *Proceedings of the 2009 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'09)*, pages 211–222. ACM, 2009.

[LMP10]   Salvatore La Torre, P. Madhusudan, and Gennaro Parlato. Model-checking parameterized concurrent programs using linear interfaces. In *CAV '10: Proc. 22nd Intl. Conf. on Computer Aided Verification*, pages 629–644. Springer, 2010.

[LN11]   Salvatore La Torre and Margherita Napoli. Reachability of multistack pushdown systems with scope-bounded matching relations. In *CONCUR*, volume 6901 of *Lecture Notes in Computer Science*, pages 203–218. Springer, 2011.

[LR08]   A. Lal and T.W. Reps. Reducing concurrent analysis under a context bound to sequential analysis. In *CAV*, volume 5123 of *LNCS*, pages 37–51. Springer, 2008.

[LS98]   D. Lugiez and Ph. Schnoebelen. The regular viewpoint on pa-processes. In *CONCUR*, volume 1466 of *LNCS*, pages 50–66. Springer, 1998.

[MP11]   P. Madhusudan and Gennaro Parlato. The tree width of auxiliary storage. In Thomas Ball and Mooly Sagiv, editors, *POPL*, pages 283–294. ACM, 2011.

[Pnu77]   Amir Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57. IEEE, 1977.

[QR05]   S. Qadeer and J. Rehof. Context-bounded model checking of concurrent software. In *TACAS*, volume 3440 of *LNCS*, pages 93–107. Springer, 2005.

[Ram00]   G. Ramalingam. Context-sensitive synchronization-sensitive analysis is undecidable. *ACM Trans. Program. Lang. Syst.*, 22(2):416–430, 2000.

[RSJ03]   T.W. Reps, S. Schwoon, and S. Jha. Weighted pushdown systems and their application to interprocedural dataflow analysis. In *SAS*, volume 2694 of *LNCS*, pages 189–213. Springer, 2003.

[Sch02]   Stefan Schwoon. *Model-Checking Pushdown Systems*. PhD thesis, Technische Universität München, 2002.

[Set10]   A. Seth. Global reachability in bounded phase multi-stack pushdown systems. In *CAV'10*, LNCS, 2010.

[SV06]   K. Sen and M. Viswanathan. Model checking multithreaded programs with asynchronous atomic methods. In *CAV*, pages 300–314. LNCS 4144, 2006.

[TMP07]   S. La Torre, P. Madhusudan, and G. Parlato. A robust class of context-sensitive languages. In *Proceedings of LICS*, pages 161–170. IEEE, 2007.

[TMP08]   S. La Torre, P. Madhusudan, and G. Parlato. Context-bounded analysis of concurrent queue systems. In *Proceedings of TACAS'08*, LNCS, pages 299–314. Springer, 2008.

[TMP09]   Salvatore La Torre, P. Madhusudan, and Gennaro Parlato. Reducing context-bounded concurrent reachability to sequential reachability. In *CAV*, volume 5643 of *LNCS*, pages 477–492. Springer, 2009.

[Var88]   Moshe Y. Vardi. A temporal fixpoint calculus. In *POPL*, pages 250–259, 1988.

[Var95]   Moshe Y. Vardi. Alternating automata and program verification. In *Computer Science Today*, volume 1000 of *Lecture Notes in Computer Science*, pages 471–485. Springer, 1995.

[VW86]   Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification (preliminary report). In *LICS*, pages 332–344. IEEE Computer Society, 1986.