# Global Model Checking of Ordered Multi-Pushdown Systems

**Mohamed Faouzi Atig**

**Uppsala University, Sweden**
mohamed_faouzi.atig@it.uu.se

**Abstract**

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

**Keywords and phrases** Concurrent Programs, Pushdown Systems, Global Model-Checking.

**Digital Object Identifier** 10.4230/LIPIcs.FSTTCS.2010.216

## 1 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., [7, 24, 15, 2, 25, 3, 13, 16]) where each thread corresponds to a sequential program with (recursive) procedure calls. In fact, it is well admitted that pushdown systems are adequate models for such kind of threads [10, 21], and therefore, it is natural to model recursive concurrent programs as multi-stack systems.

In general, multi-stack systems are Turing powerful and hence come along with undecidability of basic decision problems [20]. 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 [17, 11, 8, 22, 14].

Context-bounding has been proposed in [19] as a suitable technique for the analysis of multi-stack systems. The idea is to consider only runs of the system that can be divided into a given number of contexts, where in each context pop and push operations 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 [24], 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 systems* (BVMPS) where only those runs are taken into consideration that can be split into a given number of phases, where each phase admits pop operations of one particular stack only. In the above

IARCS Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010).
Editors: Kamal Lodaya, Meena Mahajan; pp. 216–227

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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 operations. In [9], Breveglieri et al. define *ordered multi-pushdown systems* (OMPS), which impose a linear ordering on stacks. Stack operations are constrained in such a way that a pop operation is reserved to the first non-empty stack. In [1], we show that the emptiness problem for OMPS is in 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 a complex encoding of OMPS into some class of grammars for which the emptiness problem is decidable. Moreover, we prove that the class of ordered multi-pushdown systems with $2k$ stacks are strictly more expressive than bounded-phase visibly multi-stack pushdown systems with $k$ phases.

In this paper, we consider the problem of verifying ordered multi-pushdown systems with respect to a given $w$-regular property (expressible in the linear-time temporal logics [18] or the linear-time $\mu$-calculus [26]). In particular, we are interested in solving the global model checking for ordered multi-pushdown systems 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 system. For that, we consider the class of recognizable sets of configurations defined using finite state automata [19, 2, 23].

Then, we show that for an ordered multi-pushdown system $\mathcal{M}$ the set of all predecessors $Pre^*(C)$ of a recognizable set of configurations $C$ is an effectively constructible recognizable set. The proof of this result is done by induction on the number of stacks of $\mathcal{M}$. Technically, we use a result given in [4] establishing that the set of configurations $C_n$, where the first $(n-1)$ stacks are empty, from which $\mathcal{M}$ can reach a configuration in $C$ is recognizable and effectively constructible. Then, to compute the intermediary configurations in $Pre^*(C)$ when the first $(n-1)$ stacks are not empty, we construct an ordered multi-pushdown system $\mathcal{M}'$ with $(n-1)$ stacks that: (1) performs the same operations on its stacks as the ones performed by $\mathcal{M}$ on its first $(n-1)$ stacks, and (2) simulates a push operation of $\mathcal{M}$ over its $n$-th 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 system satisfying a given $w$-regular property is recognizable and effectively constructible. Our approach also allows to obtain an 2ETIME upper bound for the model checking problem of $w$-regular properties for ordered multi-pushdown systems (matching its lower-bound [1]).

**Related works:** In [23], A. Seth shows that the set of predecessors of a recognizable set of configurations of a bounded-phase visibly multi-stack pushdown system is recognizable and effectively constructible. In fact, our results generalize the obtained result in [23] since any bounded-phase visibly multi-stack pushdown system with $k$ phases can be simulated by an ordered multi-pushdown system with $2k$ stacks [1].

To the best of our knowledge, this is the first work that addresses the global model checking for ordered multi-pushdown systems.

## 2 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 $\Sigma$ be a finite alphabet. We denote by $\Sigma^*$ (resp. $\Sigma^+$) the set of all words (resp. non empty words) over $\Sigma$, and by $\epsilon$ the empty word. A language is a (possibly infinite) set of words. We use $\Sigma_\epsilon$ to denote the set $\Sigma \cup \{\epsilon\}$.

Let $u$ be a word over $\Sigma$. The length of $u$ is denoted by $|u|$. For every $j \in [1, |u|]$, we use $u(j)$ to denote the $j^{th}$ 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) $\Sigma$ 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 $\rho$ 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 $\rho$ 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 $c \overset{u}{\underset{n}{\Longrightarrow}}_{\mathcal{T}} c'$ if one of the following two cases holds: (1) $n = 0$, $c = c'$, and $u = \epsilon$, and (2) there is a run $\rho$ 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 $c \overset{u}{\underset{n}{\Longrightarrow}}_{\mathcal{T}} c'$ 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 $Pre^+_{\mathcal{T}} = Pre_{\mathcal{T}} \circ Pre^*_{\mathcal{T}}$.

**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) $\Sigma$ 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 $\Delta$ 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 $|\mathcal{A}| = (|Q| + |\Sigma|)$. 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 $L(\mathcal{A}) = Traces_{\mathcal{T}(\mathcal{A})}(I, F)$.

## 3 Ordered Multi-Pushdown Systems

In this section, we first recall the definition of *multi-pushdown systems*. Then *ordered multi-pushdown systems* [9, 1] appear as a special case of multi-pushdown systems.

## 3.1 Multi-pushdown systems

Multi-pushdown systems 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 $t = \langle q, \gamma_1, \ldots, \gamma_n \rangle \rightarrow \langle q', \alpha_1, \ldots, \alpha_n \rangle$. 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, the system moves to the successor configuration $(q', \alpha_1 w'_1, \ldots, \alpha_n w'_n)$.

▶ **Definition 1.** A *multi-pushdown system* (MPS) is a tuple $\mathcal{M} = (n, Q, \Gamma, \Delta)$ where $n \geq 1$ is the number of stacks, $Q$ is the finite set of *states*, $\Gamma$ is the *stack alphabet* containing the special stack symbol $\perp$, and $\Delta \subseteq \big(Q \times (\Gamma_\epsilon)^n\big) \times \big(Q \times (\Gamma^*)^n\big)$ is the *transition relation* such that, for all $((q, \gamma_1, \ldots, \gamma_n), (q', \alpha_1, \ldots, \alpha_n)) \in \Delta$ 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 $\alpha'_i \in \Gamma_\epsilon$.

In the rest of this paper, we use $\langle q, \gamma_1, \ldots, \gamma_n \rangle \rightarrow_{\mathcal{M}} \langle q', \alpha_1, \ldots, \alpha_n \rangle$ to denote that the transition $((q, \gamma_1, \ldots, \gamma_n), (q', \alpha_1, \ldots, \alpha_n))$ is in $\Delta$. The size of $\mathcal{M}$, denoted by $|\mathcal{M}|$, is defined by $(n + |Q| + |\Sigma| + |\Gamma|)$.

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

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

## 3.2 Symbolic representation of MPS configurations

We show in this section how we can symbolically represent infinite sets of MPS configurations using special kind of finite automata which were introduced in [23]. Let $\mathcal{M} = (n, Q, \Gamma, \Delta)$ be a MPS. A $\mathcal{M}$-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 $w_1, \ldots, w_n \in \mathit{Stack}(\mathcal{M})$ 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 $\mathcal{M}$-automaton is closed under boolean operations and that the emptiness and membership problems are decidable in polynomial time.

## 3.3 Ordered multi-pushdown systems

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

▶ **Definition 2.** An ordered multi-pushdown system (OMPS for short) is a multi-pushdown system $(n, Q, \Gamma, \Delta)$ such that $\Delta$ contains only the following types of transitions:

- $\langle q, \gamma, \epsilon, \ldots, \epsilon\rangle \rightarrow_{\mathcal{M}} \langle q', \gamma''\gamma', \epsilon, \ldots, \epsilon\rangle$ for some $q, q' \in Q$ and $\gamma, \gamma', \gamma'' \in (\Gamma \setminus \{\bot\})$.
- $\langle q, \gamma, \epsilon, \ldots, \epsilon\rangle \rightarrow_{\mathcal{M}} \langle q', \epsilon, \ldots, \epsilon, \gamma', \epsilon, \ldots, \epsilon\rangle$ for some $q, q' \in Q$ and $\gamma, \gamma' \in (\Gamma \setminus \{\bot\})$ ($\gamma'$ is pushed on one of stacks 2 to $n$).
- $\langle q, \bot, \ldots, \bot, \gamma, \epsilon, \ldots, \epsilon\rangle \rightarrow_{\mathcal{M}} \langle q', \gamma'\bot, \bot, \ldots, \bot, \epsilon, \ldots, \epsilon\rangle$ for some $q, q' \in Q$ and $\gamma, \gamma' \in (\Gamma \setminus \{\bot\})$ ($\gamma$ is popped from one of the stacks 2 to $n$).
- $\langle q, \gamma, \epsilon, \ldots, \epsilon\rangle \rightarrow_{\mathcal{M}} \langle q', \epsilon, \ldots, \epsilon\rangle$ for some $q, q' \in Q$ and $\gamma \in (\Gamma \setminus \{\bot\})$.

For $n \geq 1$, we call a MPS (resp. OMPS) a $n$-MPS (resp. $n$-OMPS) if its number of stacks is equal to $n$.

## 4 Computing the set of predecessors for an OMPS

In this section, we show that the set of predecessors of a recognizable set $C$ of configurations of an OMPS is recognizable and effectively constructible (see Corollary 8). To simplify the presentation, we can assune without loss of generality that the set $C$ contains only one configuration of the form $(q_f, \bot, \ldots, \bot)$ where all the stacks are empty. This result is established by Lemma 3.

▶ **Lemma 3.** *Let $\mathcal{M} = (n, Q, \Gamma, \Delta)$ be an OMPS and $\mathcal{A}$ be a $\mathcal{M}$-automaton. Then, it is possible to construct, in time and space polynomial in $(|\mathcal{M}| + |\mathcal{A}|)$, an OMPS $\mathcal{M}' = (n, Q' \cup \{q_f\}, \Gamma, \Delta')$ where $Q \subseteq Q'$, $q_f \notin Q'$, and $|\mathcal{M}'| = O(|\mathcal{M}| + |\mathcal{A}|)$ 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 $c \in Pre^*_{\mathcal{T}(\mathcal{M}')}(\{(q_f, \bot, \ldots, \bot)\})$.*

**Proof.** The proof is similar to the case of standard pushdown systems. Technically, this can be done by adding to the OMPS $\mathcal{M}$ pop rules that check, in nondeterministic way, if the current configurations belongs to $L_{\mathcal{M}}(\mathcal{A})$ by simulating the finite state automaton $\mathcal{A}$. ◀

In the following, we recall a result given in [4] establishing that the set of configurations $C'$ with empty first $(n-1)$ stacks (i.e., $C' \subseteq Q \times (\{\bot\})^{n-1} \times Stack(\mathcal{M})$) from which the OMPS $\mathcal{M}$ can reach a configuration of the form $(q, \bot, \ldots, \bot)$ where all the stacks are empty is recognizable and effectively constructible.

▶ **Lemma 4.** *Let $\mathcal{M} = (n, Q, \Gamma, \Delta)$ be an OMPS and $q \in Q$ be a state. Then, it is possible to construct, in time $O(|\mathcal{M}|^{2^{dn}})$ with $d$ is a constant, a $\mathcal{M}$-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.** To prove Lemma 4 in [4], we have defined the class of *effective generalized pushdown systems* (EGPS) where operations 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, assuming that $L$ is in some class of languages for which checking whether $L$ intersects regular languages is decidable. We have shown in [4] that the automata-based saturation procedure for computing the set of predecessors in standard pushdown systems [5] can be extended to prove that for EGPS too the set of all predecessors of a recognizable set of configurations is an effectively constructible recognizable set.

Then, we have shown that, given an OMPS $\mathcal{M}$ with $n$ stacks, it is possible to construct an EGPS $\mathcal{P}$, whose pushed languages are defined by OMPSs 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 $n$-th stack of $\mathcal{M}$ when its first $(n-1)$ stacks are empty. Thus, the saturation procedure for EGPS can be used to show that Lemma 4 holds. ◀

Next, we state our main theorem which is a generalization of the result obtained in [23].

▶ **Theorem 5.** *Let $\mathcal{M} = (n, Q, \Gamma, \Delta)$ be an OMPS 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.** We proceed by induction on the number of stacks of the OMPS $\mathcal{M}$.

**Basis.** $n = 1$. Then, $\mathcal{M}$ is a pushdown system. From [5], we know that the emptiness problem for $\mathcal{M}$ can be solved in time polynomial in $|\mathcal{M}|$.

**Step.** $n > 1$. Then, we can use Lemma 4 to construct, in time $O(|\mathcal{M}|^{2^{d'n}})$ with $d'$ is a constant (we assume w.l.o.g that $d' < d$), a $\mathcal{M}$-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 $\tau' \in \Delta^*$. Afterwards, we assume w.l.o.g that the $\mathcal{M}$-automaton has no $\epsilon$-transitions.

Let $\mathcal{M}_{[1,n[} = (n, Q, \Gamma, \Delta_{[1,n[})$ be the OMPS built from $\mathcal{M}$ by discarding the set of pop operations of $\mathcal{M}$ over the $n^{th}$ stack. Formally, we have $\Delta_{[1,n[} = \Delta \cap \big((Q \times (\Gamma_\epsilon)^{n-1} \times \{\epsilon\}) \times (Q \times (\Gamma^*)^n)\big)$. 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 $q'' \in Q$, $w \in Stack(\mathcal{M})$, $\tau' \in \Delta^*$, and $\tau \in \Delta^*_{[1,n[}$ 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 OMPS $\mathcal{M}_{[1,n[}$ can only have push operations over its $n$-th 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 $w = vw_n$ and $(q', w_1, \ldots, w_{n-1}, \bot) \overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q'', \bot, \ldots, \bot, v)$.

On the other hand, let $\mathcal{M}' = (n-1, Q \times Q_{\mathcal{A}'}, \Gamma, \Delta')$ be an $(n-1)$-OMPS built up from the OMPS $\mathcal{M}_{[1,n[}$ and the FSA $\mathcal{A}'$ such that $\langle (q_1, p_1), \gamma_1, \ldots, \gamma_{n-1}\rangle \rightarrow_{\mathcal{M}'} \langle (q_2, p_2), \alpha_1, \ldots, \alpha_{n-1}\rangle$ if and only if $\langle q_1, \gamma_1, \ldots, \gamma_{n-1}, \epsilon\rangle \rightarrow_{\mathcal{M}_{[1,n[}} \langle q_2, \alpha_1, \ldots, \alpha_{n-1}, \alpha_n\rangle$ and $p_2 \overset{\alpha_n}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p_1$ for some $\alpha_n \in \big((\Gamma \setminus \{\bot\}) \cup \{\epsilon\}\big)$. In fact, the OMPS $\mathcal{M}'$ defines a kind of synchronous product between the pushed word over the $n$-th stack of OMPS $\mathcal{M}_{[1,n[}$ and the reverse of the input word of the FSA $\mathcal{A}'$. Observe that the size of the constructed $(n-1)$- OMPS $\mathcal{M}'$ is $O(|\mathcal{M}|^2)$).

Then, the relation between $\mathcal{M}'$, $\mathcal{M}_{[1,n[}$, and $\mathcal{A}'$ is given by Lemma 6 which follows immediately from the definition of $\mathcal{M}'$.

▶ **Lemma 6.** *$((q_1, p_1), w_1, \ldots, w_{n-1}) \overset{\varsigma}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}')} ((q_2, p_2), \bot, \ldots, \bot)$ if and only if there is a $v \in (\Gamma \setminus \{\bot\})^*$ such that $(q_1, w_1, \ldots, w_{n-1}, \bot) \overset{\tau}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q_2, \bot, \ldots, \bot, v\bot)$ and $p_2 \overset{v}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p_1$.*

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 $O(|\mathcal{M}|^{2^{d(n-1)+2}})$, a $\mathcal{M}'$-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 $\mathcal{M}'$-automaton $\mathcal{A}_{(q'',p'')}$ and the $\mathcal{M}$-automaton $\mathcal{A}'$, we can construct a $\mathcal{M}$-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) $q'' \overset{\bot^{n-1}}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p''$, (2) $((q', p'), w_1, \ldots, w_{n-1}) \in L_{\mathcal{M}'}(\mathcal{A}_{(q'',p'')})$, and (3) $p' \overset{w_n}{\Longrightarrow}{}^*_{\mathcal{T}(\mathcal{A}')} p$ 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 7.** $L_{\mathcal{M}}(\mathcal{A}) = Pre^*_{\mathcal{T}(\mathcal{M})}(\{(q, \bot, \ldots, \bot)\})$.

**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) $q'' \xRightarrow{\perp^{n-1}}{}^*_{\mathcal{T}(\mathcal{A}')} p''$, (2) $((q', p'), w_1, \ldots, w_{n-1}) \in L_{\mathcal{M}'}(\mathcal{A}_{(q'',p'')})$, and (3) $p' \xRightarrow{w_n}{}^*_{\mathcal{T}(\mathcal{A}')} p$ for some $p \in F_{\mathcal{A}'}$.

So, we can apply Lemma 6 to the run $((q', p'), w_1, \ldots, w_{n-1}) \xRightarrow{\varsigma}{}^*_{\mathcal{T}(\mathcal{M}')} ((q'', p''), \perp, \ldots, \perp)$ to show that there is $v \in \Gamma^*$ such that $(q', w_1, \ldots, w_{n-1}, \perp) \xRightarrow{\tau}{}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q'', \perp, \ldots, \perp, v)$ and $p'' \xRightarrow{v}{}^*_{\mathcal{T}(\mathcal{A}')} p'$. Thus, we have $(q', w_1, \ldots, w_{n-1}, w_n) \xRightarrow{\tau}{}^*_{\mathcal{T}(\mathcal{M})} (q'', \perp, \ldots, \perp, vw_n)$.

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

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

($\supseteq$) Let $(q', w_1, \ldots, w_n) \in Pre^*_{\mathcal{T}(\mathcal{M})}(\{(q, \perp, \ldots, \perp)\})$. Then, there are $q'' \in Q$, $v \in \Gamma^*$, $\tau' \in \Delta^*$, and $\tau \in \Delta^*_{[1,n[}$ such that:

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

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

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

Then, we can apply Lemma 6 to $(q', w_1, \ldots, w_{n-1}, \perp) \xRightarrow{\tau}{}^*_{\mathcal{T}(\mathcal{M}_{[1,n[})} (q'', \perp, \ldots, \perp, v)$ and $p'' \xRightarrow{v}{}^*_{\mathcal{T}(\mathcal{A}')} p'$ to show that $((q', p'), w_1, \ldots, w_{n-1}) \xRightarrow{\varsigma}{}^*_{\mathcal{T}(\mathcal{M}')} ((q'', p''), \perp, \ldots, \perp)$. 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 $q'' \xRightarrow{\perp^{n-1}}{}^*_{\mathcal{T}(\mathcal{A}')} p''$, $((q', p'), w_1, \ldots, w_{n-1}) \in L_{\mathcal{M}'}(\mathcal{A}_{(q'',p'')})$, and $p' \xRightarrow{w_n}{}^*_{\mathcal{T}(\mathcal{A}')} p$ 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 Lemma 7. ◀

This terminates the proof of Theorem 5. ◀

As an immediate consequence of Theorem 5 and Lemma 3, we obtain:

▶ **Theorem 8.** *Let $\mathcal{M} = (n, Q, \Gamma, \Delta)$ be an OMPS and $\mathcal{A}'$ be a $\mathcal{M}$-automaton. Then, it is possible to construct, in time $O((|\mathcal{M}| + |\mathcal{A}'|)^{2^{dn}})$ where $d$ is a constant, a $\mathcal{M}$-automaton $\mathcal{A}$ such that $|\mathcal{A}| = O((|\mathcal{M}| + |\mathcal{A}'|)^{2^{dn}})$ 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 $Pre^+$ preservers also recognizability.

▶ **Theorem 9.** *Let $\mathcal{M} = (n, Q, \Gamma, \Delta)$ be an OMPS and $\mathcal{A}'$ be a $\mathcal{M}$-automaton. Then, it is possible to construct, in time $O((|\mathcal{M}| + |\mathcal{A}'|)^{2^{dn}})$ where $d$ is a constant, a $\mathcal{M}$-automaton $\mathcal{A}$ such that $|\mathcal{A}| = O((|\mathcal{M}| + |\mathcal{A}'|)^{2^{dn}})$ and $L_{\mathcal{M}}(\mathcal{A}) = Pre^+_{\mathcal{T}(\mathcal{M})}(L_{\mathcal{M}}(\mathcal{A}'))$.*

**Proof.** For $Pre^+$, it is sufficient to construct a $\mathcal{M}$-automaton that recognizes the set $Pre_{\mathcal{T}(\mathcal{M})}(L_{\mathcal{M}}(\mathcal{A}))$ which is an easy extension of the construction given in [6] for standard pushdown systems. ◀

# 5 Applications to Linear-Time Global Model Checking

## 5.1 The repeated state global reachability problem

Let $\mathcal{M} = (n, Q, \Gamma, \Delta)$ be an ordered multi-pushdown system. In the this section, we are interested in solving *the repeated state global 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 $\mathcal{M}_{[1,i]} = (n, Q, \Gamma, \Delta_{[1,i]})$ the OMPS built from $\mathcal{M}$ by discarding pop operations of $\mathcal{M}$ over the last $(n-i)$ stacks. Formally, we have $\Delta_{[1,i]} = \Delta \cap \big((Q \times (\Gamma_\epsilon)^i \times (\{\epsilon\})^{n-i}) \times (Q \times (\Gamma^*)^n)\big)$.

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 $(q, w_1, \ldots, w_n) \in \mathit{Conf}(\mathcal{M})$ such that $w_1 = \cdots = w_{i-1} = \bot$ and $w_i = \gamma u$ for some $u \in \mathit{Stack}(\mathcal{M})$. 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 $j \neq i$. Then, the solution of the *repeated state global reachability problem* is based on the following fact:

▶ **Theorem 10.** *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 is $i \in [1, n]$, $q \in Q$, and $\gamma \in \Gamma$ such that:*

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

**Proof.** ($\Rightarrow$) : Let $\rho = c_0 t_0 c_1 t_1 c_2 t_2 \cdots$ be an infinite run of $\mathcal{T}(\mathcal{M})$ starting from $c_0 = c$. Recall that 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 $c_j \xrightarrow{t_j}_{\mathcal{T}(\mathcal{M})} c_{j+1}$. 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 $i$-th stack of $\mathcal{M}$. Hence, 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 $d_h$-th stack of $\mathcal{M}$. This implies that for every $h \geq r$, we have $c_h \xrightarrow{t_h}_{\mathcal{T}(\mathcal{M}_{[1,i]})} c_{h+1}$. Moreover, we must have $c_{k_j}$ is in $Q \times (\{\bot\})^{i-1} \times ((\Gamma \setminus \{\bot\})^* \cdot \mathit{Stack}(\mathcal{M})) \times (\mathit{Stack}(\mathcal{M}))^{n-i}$ since $t_{k_j}$ is a pop operations over the $i$-th stack of $\mathcal{M}$.

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 $\rho$ such that $j_0 \geq r$ and $t_{j_0}$ is a pop transition over the $i$-stack of $\mathcal{M}$, for every $\ell > 0$, $c_{j_\ell}$ is the first configuration of $\rho$ such that $j_\ell > j_{\ell-1}$ and $t_{j_\ell}$ 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 $Q \times (\{\bot\})^{i-1} \times ((\Gamma \setminus \{\bot\})^* \cdot \mathit{Stack}(\mathcal{M})) \times (\mathit{Stack}(\mathcal{M}))^{n-i}$.

Now, for every $l \geq 0$, let $\pi^{(l)}$ be the suffix of $\pi$ 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 $i$-th stack.

Construct a subsequence $\pi' = c_{z_0} c_{z_1} c_{z_2} \cdots$ of $\pi$ as follows: $c_{z_0}$ is the first configuration of $\pi$ 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 $\pi'$ whose elements have all the same state $q$, and the same symbol $\gamma$ on the top of the $i$-th stack. Observe that $c_{x_0}, c_{x_1}, c_{x_2}, \ldots$ are in $C_i^{(q,\gamma)}$.

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

$$c_0 \xrightarrow{\tau}{}^*_{\mathcal{T}(\mathcal{M})} c_{x_0} \xrightarrow{\tau'}{}^+_{\mathcal{T}(\mathcal{M})} c_{q_f} \xrightarrow{\tau''}{}^*_{\mathcal{T}(\mathcal{M})} c_{x_b}$$

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 $\pi$ (and so, $\pi'$ and $\pi''$), we have

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

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 $\pi'$ and $\pi''$ all the configurations of $\rho$ 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:

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

and

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

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

($\Leftarrow$) : It is easy to see that we can use (1) and (2) to construct a run starting from $c$ that visits infinitely often the state $q_f$.

◀

Since the sets of configurations $C_i^{(q,\gamma)}$ and $(\{q_f\} \times (Stack(\mathcal{M}))^n)$ are recognizable, we can use Theorem 8 and Theorem 9 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 11.** *Let $\mathcal{M} = (n, Q, \Gamma, \Delta)$ be an OMPS 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 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 infinitely often the state $q$.*

## 5.2 $w$-regular properties

In this section, we assume that the reader is familiar with $w$-regular properties expressed in the linear-time temporal logics [18] or the linear time $\mu$-calculus [26]. For more details, the reader is referred to [18, 28, 26, 27].

Let $\varphi$ be an $w$-regular formula built from a set of atomic propositions $Prop$, and let $\mathcal{M} = (n, Q, \Gamma, \Delta)$ be an OMPS 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. In the following, 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 the formula $\varphi$.

To solve this problem, we adopt an approach similar to [6, 5] and we construct a Buchi automaton $\mathcal{B}_{\neg\varphi}$ over the alphabet $2^{Prop}$ accepting the negation of $\varphi$ [28, 27]. Then, we compute the product of the OMPS $\mathcal{M}$ and of the Büchi automaton $\mathcal{B}_{\neg\varphi}$ to obtain an $n$-OMPS $\mathcal{M}_{\neg\varphi}$ with a set of repeating states $G$. Now, it is easy to see that the original problem can be reduced to the *repeated state global 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 11, we obtain:

▶ **Theorem 12.** *Let $\mathcal{M} = (n, Q, \Gamma, \Delta)$ be an OMPS with a labeling function $\Lambda$, and let $\varphi$ be a linear time $\mu$-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, a $\mathcal{M}$-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 $c$ that does not satisfy $\varphi$.*

**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 $\neg\varphi$ having exponential size in $|\varphi|$ [28, 26]. Therefore, the product of $\mathcal{M}$ and $\mathcal{B}_{\neg\varphi}$ has polynomial size in $|\mathcal{M}|$ and exponential size in $|\varphi|$. Applying Theorem 11 to the $n$-OMPS $\mathcal{M}_{\neg\varphi}$ (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 a $\mathcal{M}$-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 $\varphi$ 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 $\varphi$.

▶ **Theorem 13.** *The model checking problem for the linear-time temporal logics or the linear-time $\mu$-calculus and OMPSs is 2ETIME-complete.*

**Proof.** The 2ETIME upper bound is established by Theorem 12. To prove hardness, we use the fact that the emptiness problem for ordered multi-pushdown automata is 2ETIME-complete [1]. ◀

## 6 Conclusion

We have shown that the set of all predecessors of a recognizable set of configurations of an ordered multi-pushdown system is an effectively constructible recognizable set. We have also proved that the set of all configurations of an ordered multi-pushdown system 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 [5, 12] for standard pushdown systems.

**Acknowledgements** I want to thank Ahmed Bouajjani who greatly helped by reading this paper at various stages.

## References

1 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.

2 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.

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

4 Mohamed Faouzi Atig. From multi to single stack automata. In *CONCUR*, volume 6269 of *Lecture Notes in Computer Science*, pages 117–131. Springer, 2010.

**5** 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.

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

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

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

**9** 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.

**10** 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.

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

**12** 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.

**13** 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.

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

**15** 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.

**16** 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.

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

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

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

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

**21** 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.

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

**23** Anil Seth. Global reachability in bounded phase multi-stack pushdown systems. In *CAV*, volume 6174 of *Lecture Notes in Computer Science*, pages 615–628. Springer, 2010.

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

**25** 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.

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

**27** 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.

**28** 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.