Information Processing Letters 116 (2016) 638–643

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

# Marking shortest paths on pushdown graphs does not preserve MSO decidability

Arnaud Carayol [a], Olivier Serre [b,*]

[a] LIGM (CNRS & Université Paris Est), France
[b] LIAFA (CNRS & Université Paris Diderot – Paris 7), France

ARTICLE INFO

*Article history:*
Received 14 October 2015
Received in revised form 10 March 2016
Accepted 11 April 2016
Available online 30 May 2016
Communicated by Krishnendu Chatterjee

*Keywords:*
Formal methods
MSO logic
Pushdown graphs

ABSTRACT

In this paper we consider pushdown graphs, *i.e.* infinite graphs that can be described as transition graphs of deterministic real-time pushdown automata. We consider the case where some vertices are designated as being final and we build, in a breadth-first manner, a marking of edges that lead to such vertices (*i.e.*, for every vertex that can reach a final one, we mark all out-going edges laying on some shortest path to a final vertex).

Our main result is that the edge-marked version of a pushdown graph may itself no longer be a pushdown graph, as we prove that the MSO theory of this enriched graph may be undecidable.



## 1. Introduction

The original motivation of this paper comes from the following work plan: design algorithms working on *infinite* graphs and computing classical objects from graph theory. One firstly targeted set of algorithms are naturally those computing spanning trees, *e.g.* spanning trees built by performing a breadth-first search or the ones built by performing a depth-first search. In particular, breadth-first search seems to be a good candidate as it can easily be defined by a smallest fixpoint computation.

Of course, to ensure termination one has to identify reasonable classes of infinite graphs: obviously such a class should provide finite description of its elements and the graphs should have some good decidability properties. The simplest such class is the class of transition graphs of pushdown automata: they are finitely described by the underlying pushdown automata and they enjoy many good properties, in particular with respect to logic and games (see *e.g.* [2,5]). In particular, monadic second-order logic (MSO) is decidable for any pushdown graph.

Another expected property of our algorithm is that it should be reflective[1] in the following sense: the produced outputs should belong to the same class of structures as the inputs. Equivalently, in the setting of pushdown graphs, it means that we want to design an algorithm that takes as an input a pushdown graph and produces as an output another pushdown graph that is an isomorphic copy of the input graph enriched with a marking of some edges that corresponds to a breadth-first search spanning tree.

The main result of this paper is that such an algorithm does not exist, *i.e.* there is no algorithm that takes as an input a pushdown graph and returns a copy of it marked with a breadth-first search spanning tree. The roadmap to prove this result is to exhibit a pushdown graph such that when marked with a breadth-first search spanning

[1] In programming languages, reflection is the process by which a computer program can observe and dynamically modify its own structure and behaviour. See [1] for an example of reflection in the richer setting of collapsible pushdown automata and recursion schemes.

* Corresponding author.
*E-mail address:* olivier.serre@cnrs.fr (O. Serre).

http://dx.doi.org/10.1016/j.ipl.2016.04.015
0020-0190/

tree leads to a graph with an undecidable MSO theory: as pushdown graphs enjoy decidable MSO theories it directly permits to conclude.

The paper starts by introducing in Section 2 the classical objects and formally defines the problem under study. Our main results are proven in Section 3 while we briefly discuss some consequences in Section 4.

## 2. Preliminaries

An ***alphabet*** $A$ is a finite set of letters. In the sequel $A^*$ denotes the set of finite words over $A$ and the ***empty word*** is written $\varepsilon$. The length of a word $u$ is denoted by $|u|$ and for any $k \geq 0$, we let $A^{\leq k} = \{u \mid |u| \leq k\}$. Let $u$ and $v$ be two finite words. Then $u \cdot v$ (or simply $uv$) denotes the ***concatenation*** of $u$ and $v$.

Let $A$ be an alphabet. An $A$-labeled (oriented) ***graph*** is a pair $G = (V, E)$ where $V$ is a (possibly infinite) set of vertices and $E \subseteq V \times A \times V$ is a (possibly infinite) set of edges. In the sequel we write $v \xrightarrow{a} v'$ to denote that $(v, a, v') \in E$.

A vertex $v'$ is ***reachable*** from a vertex $v$ if there is a sequence $v_1, \ldots, v_\ell$ of vertices together with a sequence of letters $a_1, \ldots, a_{\ell-1}$ such that $v_1 = v$, $v_\ell = v'$ and $v_i \xrightarrow{a_i} v_{i+1}$ for every $i = 1, \ldots, \ell - 1$.

### 2.1. Pushdown graphs

A ***deterministic real-time pushdown automaton*** is defined as a tuple $\mathcal{P} = (Q, A, \Gamma, \bot, q_{in}, Q_{fin}, \delta)$ where $Q$ is a finite set of control states, $A$ is a finite input alphabet, $\Gamma$ is a finite stack alphabet, $\bot \in \Gamma$ is a bottom-of-stack symbol, $q_{in} \in Q$ is an initial state, $Q_{fin} \subseteq Q$ is a set of final states and $\delta : Q \times \Gamma \times A \to Q \times \Gamma^{\leq 2}$ is a *partial* transition function such that

- $\delta(q, \bot, a) = (q', u) \Rightarrow u \in (\Gamma \setminus \{\bot\})\bot \cup \{\bot\}$, *i.e.* $\bot$ cannot be removed.
- $\delta(q, \gamma, a) = (q', u)$ and $\gamma \neq \bot \Rightarrow u \in (\Gamma \setminus \{\bot\})^{\leq 2}$, *i.e.* $\bot$ cannot be pushed.

A ***configuration*** of $\mathcal{P}$ is a pair $(q, \sigma) \in Q \times (\Gamma \setminus \{\bot\})^*\bot$ consisting of a control state and a well-formed stack content. The ***initial configuration*** of $\mathcal{P}$ is $(q_{in}, \bot)$ and the ***final configurations*** of $\mathcal{P}$ are those of the form $(q_{fin}, \bot)$ with $q_{fin} \in Q_{fin}$.

Let $(q, \sigma)$ and $(q', \sigma')$ be two configurations, and let $a \in A$ be a letter. Then, there is an ***a-labelled transition*** from $(q, \sigma)$ to $(q', \sigma')$, denoted $(q, \sigma) \xrightarrow{a} (q', \sigma')$, if and only if one has $\delta(q, \gamma, a) = (q', u)$ where $\sigma = \gamma\sigma''$ and $\sigma' = u\sigma''$, *i.e.* $\sigma'$ is obtained from $\sigma$ by replacing its top symbol $\gamma$ by $u$.

The ***configuration graph*** of $\mathcal{P}$ is the $A$-labeled graph $G_{\mathcal{P}} = (V_{\mathcal{P}}, E_{\mathcal{P}})$ where $V_{\mathcal{P}}$ is the set of configurations of $\mathcal{P}$ and where $E_{\mathcal{P}}$ is the transition relation defined by $\mathcal{P}$. A graph isomorphic to a graph $G_{\mathcal{P}}$ is called a ***pushdown graph***.

**Example 1.** As a running example, consider the following pushdown automaton $\mathcal{P} = (Q, A, \Gamma, \bot, q_{in}, \{q_{fin}\}, \delta)$ where one lets $Q = \{q_{in}, q_{fin}, q_\sharp\}$, $A = \{a, b, \sharp\}$, $\Gamma = \{a, b, \bot\}$ and $\delta$ be as follows:

- $\delta(q_{in}, \gamma, a) = (q_{in}, a\gamma)$ and $\delta(q_{in}, \gamma, b) = (q_{in}, b\gamma)$: in the initial state on reading a symbol in $\{a, b\}$ it is copied on top of the stack.
- $\delta(q_{in}, \gamma, \sharp) = (q_\sharp, \gamma)$: in the initial state on reading symbol $\sharp$ the state is switched to $q_\sharp$.
- For $x \in \{a, b\}$ and $\gamma \neq \bot$, $\delta(q_\sharp, \gamma, x) = (q_\sharp, \varepsilon)$ if $\gamma = x$ and $\delta(q_\sharp, \gamma, x) = (q_\sharp, x\gamma)$ if $\gamma \neq x$; and $\delta(q_\sharp, \gamma, \sharp) = (q_\sharp, \gamma)$: in the state $q_\sharp$ an input letter $\sharp$ does not change the configuration while for an input letter in $\{a, b\}$ the top symbol is popped if it is the same as the input symbol otherwise the letter is copied on top of the stack.
- $\delta(q_\sharp, \bot, x) = (q_{fin}, \bot)$ for any $x \in A$: once the stack is emptied in the state $q_\sharp$ one goes to the state $q_{fin}$.
- $\delta(q_{fin}, \bot, x) = (q_{fin}, \bot)$ for any $x \in A$: once the configuration $(q_{fin}, \bot)$ is reached it stays in forever.

The graph $G_{\mathcal{P}}$ is depicted in Fig. 1.

We are interested in defining, in a breadth-first search manner, the set of configurations from which one can reach a final configuration. For this consider the following increasing sequence $(W_i)_{i \geq 0}$ of configurations of $\mathcal{P}$ and call its limit $W$.

- $W_0 = \{(q_{fin}, \bot) \mid q_{fin} \in Q_{fin}\}$ consists only of the final configurations.
- $W_{i+1} = W_i \cup \{(q, \sigma) \mid \exists (q', \sigma') \in W_i \text{ and } a \in A \text{ s.t. } (q, \sigma) \xrightarrow{a} (q', \sigma')\}$.

Obviously, $W$ is the set of all configurations from which a final configuration is reachable. Define for every configuration $(q, \sigma)$ its rank $rk((q, \sigma))$ to be the smallest $i$ such that $(q, \sigma) \in W_i$ when exists and to be $\infty$ otherwise.

We now define a new graph $\widetilde{G}_{\mathcal{P}}$ obtained from $G_{\mathcal{P}}$ by marking those edges that go from a configuration to one with a strictly smaller rank (equivalently that decrease the rank by 1). First we let $\widetilde{A} = A \cup \{\underline{a} \mid a \in A\}$ consists of $A$ together with a marked copy of each of its elements. Then we let $\widetilde{G}_{\mathcal{P}}$ be the $\widetilde{A}$-labelled graph $(V_{\mathcal{P}}, \widetilde{E}_{\mathcal{P}})$ where

- $((q, \sigma), a, (q', \sigma')) \in \widetilde{E}_{\mathcal{P}}$ if $((q, \sigma), a, (q', \sigma')) \in E_{\mathcal{P}}$ and $rk((q', \sigma')) \geq rk((q, \sigma))$;
- $((q, \sigma), \underline{a}, (q', \sigma')) \in \widetilde{E}_{\mathcal{P}}$ if $((q, \sigma), a, (q', \sigma')) \in E_{\mathcal{P}}$ and $rk((q', \sigma')) < rk((q, \sigma))$.

Coming back to Example 1, the graph $\widetilde{G}_{\mathcal{P}}$ is depicted in Fig. 2.

Finally, one can consider a graph built out of $G_{\mathcal{P}}$ by marking only some (but at least one) shortest paths to a final configuration (the extreme case being when the marked paths form a spanning tree). More precisely, a ***well-formed marking*** of $G_{\mathcal{P}}$ is an $\widetilde{A}$-labelled graph $G = (V_{\mathcal{P}}, E)$ such that:

- For every edge $((q, \sigma), \underline{a}, (q', \sigma')) \in E_{\mathcal{P}}$, one has either $((q, \sigma), a, (q', \sigma')) \in E$ or $((q, \sigma), \underline{a}, (q', \sigma')) \in E$.