Finiteness up to bisimilarity is decidable for pushdown processes

Petr Jančar
Techn. Univ. Ostrava, Czech Republic
petr.jancar@vsb.cz

Abstract
It is shown that it is decidable if a given configuration of a pushdown automaton (pda) with no \( \varepsilon \)-transitions is bisimulation equivalent with some unspecified finite-state process. While the semidecidability of the positive case has been long clear, it is the existence of a finite effectively verifiable witness of the negative case which is the crucial point here. The presented algorithm also uses a procedure for deciding bisimilarity between pda configurations, which is known due to Sénizergues (1998, 2005). The complexity of the procedure is non-elementary, as shown by Benedikt, Gölker, Kiefer, and Murawski (2012), but the EXPTIME-hardness (Kučera and Mayr 2002, and Srba 2002) remains the only known complexity bound for the problem.

1 Introduction

The question of deciding semantic equivalences of systems is a frequent topic in computer science. A closely related question is if a given system in a class \( C_1 \) has an equivalent system in a simpler class \( C_2 \). Pushdown automata (pda) constitute a well-known example. By “folklore” results, the language equivalence problem and the regularity problem are undecidable for (nondeterministic) pda. In the case of deterministic pda (dpda), the decidability and complexity results for regularity (see [8] and the references therein) preceded the famous decidability result for equivalence by Sénizergues (generalized in [5], as mentioned below).

In theory of concurrency, in process algebras and similar settings, a finer equivalence, namely bisimulation equivalence or bisimilarity, has emerged as a fundamental behavioural equivalence. An on-line survey of the results which study this equivalence in a specific area of process rewrite systems is maintained by J. Srba [6]. Here the most involved result shows the decidability of bisimilarity for pushdown processes, generated by (nondeterministic) pda with a restricted use of \( \varepsilon \)-rules; this was shown by Sénizergues [5] who thus generalized his above mentioned result for dpda. (Further comments on \( \varepsilon \)-rules are given at the end of the paper.) We note that there is no known upper bound on the complexity of this decidable problem; recently Benedikt, Gölker, Kiefer, and Murawski [1] showed a non-elementary lower bound. In the deterministic case, the equivalence problem is PTIME-hard, and has a non-elementary upper bound due to Stirling [7].

In the light of the above facts, one could expect that the “regularity” in the bisimilarity setting, i.e. the question if a given pda is bisimilar with an unspecified finite-state system, is decidable as well, and that this problem might be even easier than the equivalence problem solved in [5]. Nevertheless, this decidability question has been open so far, as also indicated in [2] where some related problems are tackled (also in the area of higher-order pda).

Here we answer the question positively. The presented algorithm, answering if a given pda has a bisimilar finite-state system, uses the result of [5] as a black-box procedure. By the above mentioned result [1], the complexity of the algorithm is non-elementary; nevertheless, the EXPTIME-hardness (Kučera and Mayr 2002, and Srba 2002, see [5]) remains the only known complexity bound for this problem.

The crucial point is to show that there is always a finite effectively verifiable witness when a pda configuration is not bisimilar with any finite-state process. Such witness is in the form of a
“loop” pda computation that after some finite prefix increases the stack by repeating the use of a fixed sequence of pda rules forever. We show that there must be such a loop path that visits infinitely many bisimilarity classes, by increasing the “equivalence-level” w.r.t. the configuration that corresponds to the “limit case” of the mentioned stack increasing. The (black-box) procedure deciding equivalence is used to show a verifiable bound on the number of performing the loop that allows us to confirm the witness property of the path in finite time.

2 Definitions, result, and proof

2.1 Labelled transition systems, bisimilarity, eq-levels, non-regularity

For a set $A$, by $A^*$ we denote the set of finite sequences of elements of $A$, i.e. of words over $A$; by $\varepsilon$ we denote the empty word, and we put $A^+ = A^* \setminus \{\varepsilon\}$. By $|w|$ we denote the length of $w \in A^*$. We use $\mathbb{N}$ to denote the set of nonnegative integers $\{0, 1, 2, \ldots\}$.

**LTS.** A labelled transition system, an LTS for short, is a tuple $L = (S, \mathcal{A}, (\xrightarrow{a})_{a \in \mathcal{A}})$ where $S$ is the set of states, $\mathcal{A}$ is the set of actions and $\xrightarrow{a} \subseteq S \times S$ is the set of transitions labelled with $a \in \mathcal{A}$. We write $s \xrightarrow{a} s'$ instead of $(s, s') \in \xrightarrow{a}$. For $w \in A^*$, we define $s \xrightarrow{w} s'$ inductively: $s \xrightarrow{a} s$ if $s \xrightarrow{a}$ s' and $s' \xrightarrow{a} s''$ then $s \xrightarrow{a} s''$. By $s \xrightarrow{a} s'$ we denote that $s'$ is reachable from $s$, i.e. $s \xrightarrow{w} s'$ for some $w \in A^*$. An LTS $L = (S, \mathcal{A}, (\xrightarrow{a})_{a \in \mathcal{A}})$ is image-finite if the set $\{s' \mid s \xrightarrow{a} s'\}$ is finite for any $s \in S$ and $a \in \mathcal{A}$, and $L$ is finite-branching if the set $\{s' \mid s \xrightarrow{a} s'\}$ for some $a \in \mathcal{A}$ is finite for every $s \in S$; if both $S$ and $\mathcal{A}$ are finite then $L$ is a finite LTS.

**(Stratified) bisimilarity.** Given an LTS $L = (S, \mathcal{A}, (\xrightarrow{a})_{a \in \mathcal{A}})$, we say that $B \subseteq S \times S$ covers $(s, t) \in S \times S$ if for any $s \xrightarrow{a}$ s' there is $t \xrightarrow{a}$ t' such that $(s', t') \in B$, and for any $t \xrightarrow{a}$ t' there is $s \xrightarrow{a}$ s' such that $(s', t') \in B$. For $B, B' \subseteq S \times S$, we say that $B$ covers $B'$ if $B$ covers each $(s, t) \in B'$. A set $B \subseteq S \times S$ is a bisimulation if $B$ covers $B$. States $s, t \in S$ are bisimilar, written $s \sim t$, if there is a bisimulation $B$ containing $(s, t)$. In fact, $\sim \subseteq S \times S$ is the maximal bisimulation, the union of all bisimulations.

We put $\sim_0 = S \times S$. For $k \in \mathbb{N}$, $\sim_{k+1} \subseteq S \times S$ is the set of all pairs covered by $\sim_k$.

When comparing the states from different LTSs $L_1, L_2$, we implicitly refer to the disjoint union of $L_1$ and $L_2$. The next proposition captures some standard facts that are easy to check; see Point (3), we note that $\cap_{k \in \mathbb{N}} \sim_k$ is a bisimulation in image-finite systems.

**Proposition 1.** For any LTS $L = (S, \mathcal{A}, (\xrightarrow{a})_{a \in \mathcal{A}})$ we have:
1. $\sim_0 = S \times S$. For $k \in \mathbb{N}$, $\sim_k \subseteq S \times S$ is the set of all pairs covered by $\sim_k$.
2. $S \times S = \sim_0 \supseteq \sim_1 \supseteq \sim_2 \supseteq \cdots$, and $\cap_{k \in \mathbb{N}} \sim_k \supseteq \sim_0$.
3. If $L$ is image-finite then $\cap_{k \in \mathbb{N}} \sim_k = \sim$.
4. If $s_0 \sim_0 t_0$ and $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} \cdots$ is a finite or infinite path in $L$ then there is a matching path $t_0 \xrightarrow{a_1} t_1 \xrightarrow{a_2} t_2 \xrightarrow{a_3} \cdots$ such that $s_i \sim t_i$ for all $i = 0, 1, 2, \ldots$.

**Equivalence levels.** In image-finite systems we define the *equivalence-level*, or the *eq-level*, for each pair of states:

$$\text{EqLv}(s, t) = k \ (k \in \mathbb{N}) \text{ if } s \sim_k t \text{ and } s \not\sim_{k+1} t; \quad \text{EqLv}(s, t) = \omega \text{ if } s \sim t.$$

Here $\omega$ stands for an infinite number satisfying $n < \omega$ for all $n \in \mathbb{N}$. We do not need other infinite ordinals since we further only deal with image-finite systems (though some claims are valid in general). We now observe a simple fact (easily derivable from Prop. 1):

**Proposition 2.** For any states $r, s, t$ of an LTS, if $r \sim s$ then $\text{EqLv}(r, t) = \text{EqLv}(s, t)$.

**Bisimilarity quotient.** The bisimilarity quotient of an LTS $L = (S, \mathcal{A}, (\xrightarrow{a})_{a \in \mathcal{A}})$ is the LTS $L_/ = (\{ [s] \mid s \in S \}, \mathcal{A}, (\xrightarrow{a})_{a \in \mathcal{A}})$ where $[s] = \{s' \mid s' \sim s\}$ and $[s] \xrightarrow{a} [t]$ if $s' \xrightarrow{a} t'$ for some $s' \in [s]$ and $t' \in [t]$. (In fact, then for each $s' \in [s]$ there is $t' \in [t]$ such that $s' \xrightarrow{a} t'$.) We can easily check that $s \sim [s]$ (since $\{(s, [s]) \mid s \in S\}$ is a bisimulation in the union of $L$ and $L_/$.}
(Non-)regularity. Given an LTS $L = (S, A, (\rightarrow)_{a \in A})$ where $A$ is finite, we say that $s_0 \in S$ is finite up to bisimilarity, or regular for short, if the set $\{[s] ; s_0 \rightarrow^* s\}$ is finite; if the set $\{[s] ; s_0 \rightarrow^* s\}$ is infinite then $s_0$ is infinite up to bisimilarity, or non-regular for short. We note that $s_0$ is regular iff there is some state $f$ in a finite LTS such that $s_0 \sim f$.

**Proposition 3.** For any non-regular state $s_0$ of a finite-branching LTS $L$ there is an infinite path $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} \cdots$ where $s_i \neq s_j$ for all $i \neq j$.

**Proof.** Consider the reachability tree in $L$, where $[s_0]$ is the root and each branch is finished when visiting the same class $[s]$ twice. By König’s Lemma there is an infinite branch, and this branch has a matching path starting in $s_0$ in $L$ (recall Prop. [4]). □

For showing that $s_0$ is non-regular it is not necessary to demonstrate a path as in Prop. 3. It suffices to show that there is a bisim-infinite path starting in $s_0$, i.e. a path $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} \cdots$ where the set $\{[s_i] ; i \in \mathbb{N}\}$ is infinite. An example (of “increasing eq-levels w.r.t. a test state $t$”) is captured by the next proposition (which follows from Prop. 2):

**Proposition 4.** Given $L = (S, A, (\rightarrow)_{a \in A})$ and some fixed $t \in S$, if an infinite path $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} \cdots$ satisfies that for every $e \in \mathbb{N}$ there is $i$ such that $e < \text{EqLv}(s_i, t) < \omega$ then the path is bisim-infinite (and so is thus non-regular).

Another useful consequence of the previous facts is captured by Prop. 5; we first introduce the related notions. Given $L = (S, A, (\rightarrow)_{a \in A})$, for any $s \in S$ and $m \in \mathbb{N}$ we define $\text{Region}(s, m) \subseteq S$, which is always a finite set when $L$ is finite-branching:

$\text{Region}(s, m) = \{s' \mid s \xrightarrow{w} s' \text{ for some } w \in A^* \text{ such that } |w| \leq m\}$.

For any sets $R, T \subseteq S$, we define the following subset of $\mathbb{N} \cup \omega$:

$\text{SeqLv}(R, T) = \{e \mid e = \text{EqLv}(r, t) \text{ for some } r \in R \text{ and } t \in T\}$.

**Proposition 5.** In any LTS $L = (S, A, (\rightarrow)_{a \in A})$, if $r \sim s$ then $\text{SeqLv}(\text{Region}(r, m), T) = \text{SeqLv}(\text{Region}(s, m), T)$ for any $m \in \mathbb{N}$ and any $T \subseteq S$.

### 2.2 Pushdown processes, regularity problem, compositionality

**LTSs generated by pushdown automata.** A pushdown automaton (pda) is a tuple $M = (Q, \Gamma, A, \mathcal{R})$ of finite nonempty sets of control states, stack symbols, actions (also known as input letters), and (rewriting) rules, respectively. The rules are of the form $pX \xrightarrow{a} q\alpha$ where $p, q \in Q$, $X \in \Gamma, a \in A, \alpha \in \Gamma^*$. (We have no $\varepsilon$-rules $pX \xrightarrow{\varepsilon} q\alpha$; we add a comment on them later.)

A pda $M$ gives rise to the LTS $M = (\text{Conf}, A, (\rightarrow)_{a \in A})$ where the set of configurations is defined as $\text{Conf} = Q \times \text{RegStr}(\Gamma)$; the set $\text{RegStr}(\Gamma)$ contains the regular strings over $\Gamma$, i.e. the elements of $\Gamma^*$ (which are finite strings) and the infinite strings of the form $\alpha\beta\beta\cdots = \alpha\beta^\omega$ where $\alpha \in \Gamma^*, \beta \in \Gamma^*$. The relations $\rightarrow$ are induced by the following (deduction) rule: if $pX \xrightarrow{a} q\alpha$ is in $\mathcal{R}$ then $pX \xrightarrow{a} q\alpha\sigma$ for any $\sigma \in \text{RegStr}(\Gamma)$. We note that $M$ is finite-branching.

By a pushdown process we mean a pda configuration, i.e. a state of $M$ for a pda $M$.

**Remark.** If $p\alpha \xrightarrow{w} q\beta$ and $\alpha \in \Gamma^*$ then also $\beta \in \Gamma^*$ (since the strings in the rules in $\mathcal{R}$ are finite); the standard definitions would put $\text{Conf} = Q \times \Gamma^*$. We include the configurations of the form $q\alpha\beta^\omega$ for later technical convenience; we could avoid their use by introducing additional stack symbols and rules when needed but this might look less elegant.

The absence of $\varepsilon$-rules implies the following simple, but important, fact.

**Proposition 6.** In $M$ (for a pda $M$) we have $\text{EqLv}(q\alpha\sigma_1, q\alpha\sigma_2) \geq |\alpha|$.

**Computability of eq-levels.** We now recall a crucial theorem, due to Sénizergues [5]:

**Theorem 7.** There is an algorithm that, given a pda $M = (Q, \Gamma, A, \mathcal{R})$ and two configurations $p\sigma_1$, $q\sigma_2$ (where $\sigma_1, \sigma_2 \in \text{RegStr}(\Gamma)$), computes $\text{EqLv}(p\sigma_1, q\sigma_2)$. 

3
Proof. By [5] we have the decidability of \( \rho \sigma_1 \sim q \sigma_2 \) where \( \sigma_1, \sigma_2 \in \Gamma^* \). As already mentioned, the equivalence problem in our setting can be easily reduced to the standard setting by using additional stack symbols and rules. Once we have the decidability of \( \rho \sigma_1 \sim q \sigma_2 \), the computability of \( \text{EqLv}(\rho \sigma_1, q \sigma_2) \) is clear since the question if \( \rho \sigma_1 \sim_k q \sigma_2 \) is obviously decidable for any fixed \( k \) (using the finite-branching property of \( \mathcal{L}_M \)). \[\qed\]

Remark. The decidability proof for deterministic pda presented in [5] in the framework of first-order regular terms can be generalized to a direct proof for bisimilarity in the framework of terms as well; a version of this generalization appears only at the author’s web-page so far.

Decidability of pda-regularity. We now state the result of this paper; here it is convenient to stick to the initial configurations \( c_{in} \) with finite stack contents.

Problem \( \text{PdaRegul} \)

Instance: A pda \( \mathcal{M} = (Q, \Gamma, A, R) \), and a configuration \( c_{in} = p_0 \alpha_0 \) where \( \alpha_0 \in \Gamma^* \).

Question: Is \( c_{in} \) regular, i.e., is there some state \( f \) in a finite LTS such that \( c_{in} \sim f \) ?

Theorem 8. Problem \( \text{PdaRegul} \) is decidable.

We prove the theorem in the rest of the paper. We first note that Theorem [7] yields the semidecidability of \( \text{PdaRegul} \), since we can generate all finite LTSs systematically.

Remark. For any state \( s \) in an LTS \( \mathcal{L} \) and any state \( f \) in a finite LTS \( \mathcal{F} \) with \( k \) states we have \( s \sim f \) iff \( s \sim_k f \) and for each \( s' \) reachable from \( s \) there is \( g \) in \( \mathcal{F} \) such that \( s' \sim_k g \). (If there is no \( s' \) violating the above condition then the set \( \{ (r, g) \mid s \rightarrow^* r, r \sim_k g \} \) can be checked to be a bisimulation, since \( \sim_{k-1} = \sim_k = \sim \) on \( \mathcal{F} \).) Hence bisimilarity of a pushdown process and a finite-state one can be decided by a straightforward procedure (described in the literature), and we thus do not need Theorem [7] for establishing the semidecidability of \( \text{PdaRegul} \).

We will use Theorem [4] for proving the semidecidability of the complement of \( \text{PdaRegul} \). The main argument is given in the next section; here we recall some further useful notions, and the compositionality of bisimilarity for pushdown processes. We further assume a fixed pda \( \mathcal{M} = (Q, \Gamma, A, R) \) if not said otherwise.

Stack-segments as state-transformers. For each \( \alpha \in \Gamma^* \) we define the relation \( [\alpha] \) on \( Q \) and the relation \( \| \alpha \| \) on \( 2^Q \) as follows:

- For \( p, p' \in Q \) we put \( p[\alpha] p' \) if \( p \alpha \rightarrow^* p' \) (here \( p' = p' \varepsilon \) is an empty-stack configuration).
- For \( K, K' \subseteq Q \) we put \( K \| \alpha \| K' \) if \( K' = \{ p' \mid \exists p \in K : p[\alpha] p' \} \).

We note that \( \| \alpha \| \) is, in fact, a function of the type \( 2^Q \rightarrow 2^Q \). We could also define the relations \( [\alpha] \) inductively: \( p[\varepsilon] p \); if \( p \alpha \rightarrow q \alpha \) is in \( \mathcal{R} \) and \( q[\alpha] q' \) then \( p[A] q' \); if \( p[\alpha] p'[\beta] p'' \) then \( p[\alpha \beta] p'' \).

The next proposition is thus obvious (by a standard dynamic programming approach):

Proposition 9. There is a polynomial algorithm that, given a pda \( \mathcal{M} = (Q, \Gamma, A, R) \), computes all triples \( p[X] p' \) (where \( p, p' \in Q, X \in \Gamma \)), and an at most exponential number \( \mathcal{E} \in \mathbb{N} \) such that \( p[X] p' \) implies that \( p' \in \text{REGION}(pX, \mathcal{E}) \).

Hence \( p[\alpha] p' \) implies that \( p' \sigma \in \text{REGION}(p\sigma, [\alpha] \cdot \mathcal{E}) \) for any \( \sigma \in \text{RegStr}(\Gamma) \). There are also other obvious consequences; e.g., \( K \| \alpha \| K' \| \beta \| K'' \) implies \( K \| [\alpha \beta] \| K'' \), and this suggests a polynomial procedure, given \( K \subseteq Q \) and \( \alpha \in \Gamma^* \), computes \( K' \) such that \( K \| \alpha \| K' \).

We note the monotonicity of \( [\alpha] \): if \( K \| \alpha \| K' \), \( L \| \alpha \| L' \), and \( K \subseteq L \) then \( K' \subseteq L' \). It is also useful to observe that if \( p \| [\alpha] \| K \), \( p \alpha \rightarrow^* q \beta \), and \( q \| [\beta] \| L \) then \( L \subseteq K \).

Compositionality of bisimilarity. We recall that a state in the LTS \( \mathcal{L}_M \) has a structure, since it is a string \( \rho \sigma \) (where \( \sigma \in \Gamma^* \) or \( \sigma = \alpha \beta \)). Bisimilarity of pushdown configurations is compositional, i.e., it has congruence properties. Such facts are stated more naturally in the framework of terms (composed from subterms) but the following formulation suffices for our aims.

Proposition 10. The bisimilarity class of \( q \alpha \sigma \) (i.e., the state \( [q \alpha \sigma] \) in \( \mathcal{L}_M \)) is determined by \( q \alpha \sigma \) and the tuple \( ([p \alpha \sigma])_{p \in K} \) where \( \{ q \} [\alpha] K \). Hence if \( \rho \sigma \sim \rho \sigma' \) for all \( p \in K \) then \( q \alpha \sigma \sim q \alpha \sigma' \).
2.3 Finite witnesses of non-regularity of pushdown processes

Let us consider a non-regular (initial) configuration \( c_{in} = p_0\alpha_0 \) of a pda \( \mathcal{M} = (Q, \Gamma, A, R) \), where \( p_0 \in Q \) and \( \alpha_0 \in \Gamma^* \); the state \( c_{in} \) in \( L_M \) is thus non-bisimilar with any state of any finite LTS.

**Stairs.** By Prop. 5 there is an infinite path \( c_{in} = c_0 \xrightarrow{a_1} \cdots \xrightarrow{a_n} c_i \in L_M \) such that \( c_i \nless c_j \) (and thus \( c_i \neq c_j \)) for all \( i \neq j \); since \( c_{in} \) is a finite string, all \( c_i \) are finite strings. The above infinite path thus has an infinite “stair subsequence” \( c_{i_1}, c_{i_2}, c_{i_3}, \ldots \) where \( i_j < m \) implies \( |c_{i_j}| < |c_m| \) (i.e., after \( c_j \), the stack height is always bigger than in \( c_i \)). We can thus fix a sequence

\[
\begin{align*}
p_0\alpha_0 \xrightarrow{w_0} q_0X_0\delta \xrightarrow{w_1} q_1X_1A_1\delta \xrightarrow{w_3} q_2X_2A_2A_1\delta \xrightarrow{w_5} q_3X_3A_3A_2A_1\delta \xrightarrow{w_7} \cdots
\end{align*}
\]

where \( X_i \in \Gamma \) and \( q_iX_1 \xrightarrow{w_i} q_{i+1}X_{i+1}A_{i+1} \) for all \( i \in \mathbb{N}; \) when putting \( c_i' = q_kX_kA_kA_{k-1} \cdots A_1\delta \) (for \( k \in \mathbb{N} \)), we have \( c_i \nless c_j \) for all \( i \neq j \). Moreover, we choose \( \mathcal{I} \) so that \( A_i \) are nonempty suffixes of the right-hand sides of the rules \( pX \xrightarrow{a} q\alpha \) in \( \mathbb{R} \).

**Remark.** In fact, our notation suggests that \( A_i, \Gamma \in \mathbb{G} \); this could be indeed achieved by a standard extension of the stack alphabet allowing to get \( |\alpha| \leq 2 \) in all rules \( pX \xrightarrow{a} q\alpha \). But formally we do not require \( A_i \in \Gamma \).

**Loop-paths.** For \( i < j \), we refer to \( \beta = A_jA_{j-1} \cdots A_i \) as to the stack-increase of the segment \( c_i' \xrightarrow{v} c_j' \) of \( \mathcal{I} \); here \( v = w_iw_{i+1} \cdots w_{j-1} \). Since \( q_iX_i \in Q \times \Gamma \), and \( Q, \Gamma \) are finite sets, we have many loop-pairs \((i, j)\), i.e. the pairs in which \( i < j \) and \( q_iX_i = q_jX_j \). Any loop-pair \((i, j)\) has the associated triple \((q, X, \beta)\) where \( qX = q_iX_i = q_jX_j \) and \( \beta \in \Gamma^+ \) is the respective stack-increase; the loop-pair also has the associated loop-path which coincides with the prefix of \( \mathcal{I} \) up to \( c_i' \) and then continues in adding the stack-increase \( \beta \in \Gamma^+ \) forever. We write this loop-path as

\[
\begin{align*}
p_0\alpha_0 \xrightarrow{v} qX\gamma \xrightarrow{w} qX\beta\gamma \xrightarrow{w} qX\beta\beta\gamma \xrightarrow{w} \cdots w \xrightarrow{w} qX\beta^i\gamma \xrightarrow{w} \cdots ,
\end{align*}
\]

where \( qX\gamma = c_i', qX\beta = c_j' \); thus \( v = w_0w_1 \cdots w_{i-1} \) and \( \gamma = A_iA_{i-1} \cdots A_1\delta \).

For any \( i \in \mathbb{N} \) we obviously have \( qX\beta^i\gamma \sim_{i} qX\beta^j\gamma \sim_{i} qX\gamma \) (recall Prop. 6). Roughly speaking, the next lemma implies that there is a computable bound \( B \) determined by \( q, X, \beta \) and \( \gamma \) such that the bisimilarity class of \( qX\beta^i\gamma \) is visited by the path \( \mathcal{I} \) either within \( B \) steps, or never; in the latter case the path is bisim-infinite (by Prop. 7). In fact, it will suffice to show just \( qX\beta^i\gamma \nless qX\beta^i\gamma \) to derive the bisim-infiniteness of the path; this is what we use for defining the non-regularity witnesses after proving Lemma 11 (Point 1 is stated just for getting a broader view.)

**Lemma 11.** For any pda \( \mathcal{M} = (Q, \Gamma, A, \mathbb{R}) \) and \( q \in Q, X \in \Gamma, \beta \in \Gamma^+ \), there is a computable bound \( B = B_{qX\beta} \in \mathbb{N} \) such that for any \( \gamma \in \Gamma^* \) we have:

1. if \( qX\beta^i\gamma \nless qX\beta^j\gamma \) for all \( i < B \) then \( qX\beta^i\gamma \nless qX\beta^j\gamma \) for all \( i \in \mathbb{N} ; \)
2. if \( qX\beta^B\gamma \nless qX\beta^\omega \) then the set \( \{ qX\beta^i\gamma; i \in \mathbb{N} \} \) is infinite.

**Proof.** For \( q, X, \beta \) we consider the sequence \( \{q\} \langle X \rangle K_0 \langle \beta \rangle K_1 \langle \beta \rangle K_2 \langle \beta \rangle \cdots \). Let \( j \geq 1 \) be the smallest index for which \( K_j \subseteq Q \) repeats, i.e., for which \( j \) is the (unique) \( b < j \) such that \( K_b = K_j \); we put \( \ell = j - b \) and note that \( b, \ell \) are computable exponentially bounded numbers. We also put \( L = K_b \cup K_{b+1} \); we thus have \( \{q\} \langle X\beta^b \rangle L \langle \beta' \rangle L, \) hence also \( \{q\} \langle X\beta^i\beta^b \rangle L, \) (Note that \( L = 0 \) implies that \( qX\beta^\omega \sim qX\beta^\omega \) for any \( \sigma.)\)

We put \( \mathcal{E}' = |X\beta^{\ell+1}| \cdot \mathcal{E} \) where \( \mathcal{E} \) is the (exponential) number from Prop. 6 and note that

\[
\begin{align*}
p\sigma \in \text{REGION}(qX\beta^i\beta^\omega, \mathcal{E}') \quad \text{for any } p \in L \text{ and any } \sigma.
\end{align*}
\]

We now recall the notation \( \text{SEQLV}(R, T) \) before Prop. 6 and we put

\[
C = \max\{ e < \omega \mid e \in \text{SEQLV}(R, T) \} \text{ where } R = \text{REGION}(qX\beta^\omega, \mathcal{E}') \text{ and } T = \{ p\beta^\omega \mid p \in L \} \}.
\]
Hence $C \in \mathbb{N}$; we stipulate $\max \emptyset = 0$. Recalling Theorem $[7]$ we easily observe that $C$ is computable. We put $B = 1 + C + b + \ell$.

Let us now assume that $i \geq B - \ell = 1 + C + b$ and $qX^\beta \not\sim qX^\beta^\omega$; in another presentation $qX^\beta^{\beta(i-\ell)} \not\sim qX^\beta^\omega$. Using compositionality (Prop. $[10]$), and recalling that $\{q\} [X^\beta^\omega] \subseteq L$, we note that there must be $p \in L$ such that $p^\beta(i-\ell) \not\sim p^\beta^\omega$; since $p^\beta(i-\ell) \sim \rho \overset{\text{def}}{=} p^\beta^\omega$, we have $C < EqLv(p^\beta(i-\ell), p^\beta^\omega) < \omega$.

By $[3]$ we have $p^\beta(i-\ell) \in \text{REGION}(qX^\beta^{\beta(i-\ell)}, \mathcal{E})$; hence $qX^\beta^{\beta(i-\ell)} \not\sim qX^\beta^\omega$, by recalling Prop. $[3]$ and the definition of $C$.

Hence the assumption $i \geq B - \ell$ and $qX^\beta \not\sim qX^\beta^\omega$ implies that $qX^\beta^{\beta(i+1)} \not\sim qX^\beta^\omega$. Therefore, if $qX^\beta^B \not\sim qX^\beta^\omega$ then $B + j \ell < EqLv(qX^\beta^{(B+j)\gamma}, qX^\beta^\omega) < \omega$ for all $j \in \mathbb{N}$; Point 2 thus follows. Moreover, if $qX^\beta \not\sim qX^\beta^\omega$ for all $i \in \{B-\ell, B-\ell+1, \ldots, B-1\}$ then $qX^\beta \not\sim qX^\beta^\omega$ for all $i \geq B - \ell$. Thus also Point 1 follows. 

**Witness.** Assuming a pda $\mathcal{M} = (Q, \Gamma, \mathcal{A}, \mathcal{R})$, we define a witness for $p_{0\alpha\sigma}$ (meaning a witness of non-regularity of $p_{0\alpha\sigma}$) as a pair of finite paths $p_{0\alpha\sigma} \xrightarrow{w} qX^\gamma$ and $qX \xrightarrow{w} qX^\beta$ for $\beta \in \Gamma^+$ and $qX^\beta \not\sim qX^\beta^\omega$ for $B = B_{qX^\beta}$ from Lemma $[11]$ the path of the form $[2]$ is then bisim-infinite.

Since $B_{qX^\beta}$ is computable, we can effectively verify if a given pair of finite paths is a witness for $p_{0\alpha\sigma}$. Hence after we prove Lemma $[12]$ the semidecidability of non-regularity of pushdown processes will be established, and a proof of Theorem $[5]$ will be finished.

**Lemma 12.** There is a witness for any non-regular pda configuration.

**Proof.** Let us consider a non-regular configuration $p_{0\alpha\sigma}$ of a pda $\mathcal{M} = (Q, \Gamma, \mathcal{A}, \mathcal{R})$; we further refer to a fixed path $[1]$, given in the following two presentations, where $c'_i = q_iX_iA_iA_{i-1}\ldots A_1\delta$:

$$p_{0\alpha\sigma} = c_m \xrightarrow{u} c'_0 \xrightarrow{w_0} c'_1 \xrightarrow{w_1} \ldots c'_i \xrightarrow{w \mathcal{A}}$$

For every $i \in \mathbb{N}$ and $d \in \{0, 1, \ldots, i\}$ we define the stamp $S(i, d)$, i.e. the stamp of $c'_i$ in depth $d$, as the pair $(qX, K)$ where $qX = q_{i-d}X_i\ldots d$ and $\{q_i\} [X_iA_iA_{i-1}\ldots A_{i-d+1}] K$. For every $i \geq |Q| \cdot |\Gamma| \cdot 2^{|Q|}$ (where $|\cdot|$ denotes the cardinality) we are guaranteed to find the smallest $d_1$ such that $S(i, d_1)$ is repeated, i.e., there is (the unique) $d_2 < d_1$ such that $S(i, d_2)$ is $S(i, d_2)$.

We put $qX = q_{i-d_1}X_{i-d_1} = q_{i-d_2}X_{i-d_2}$, $\alpha = A_iA_{i-1}\ldots A_{i-d_1+1}$, $\beta = A_iA_{i-d_2}A_{i-d_2-1}\ldots A_{i-d_1+1}$, $\gamma = A_{i-d_1}A_{i-d_1-1}\ldots A_1\delta$. Hence

$$c'_{i-d_1} = qX^\gamma, c'_{i-d_2} = qX^\beta \gamma, \text{ and } c'_i = q_iX_i\alpha\beta\gamma.$$ 

By $K$ we denote the set satisfying $\{q_i\} [X_i\alpha] K [\beta] K$. Hence $(i-d_1, i-d_2)$ is a loop-pair with the associated triple $(q, X, \beta)$; moreover, $qX \not\sim^* q_iX_i\alpha$ and thus $\{q\} [X] K_0 \supseteq K$.

In the sequence $\{q\} [X] K_0 [\beta] K_1 [\beta] K_2 [\beta] K_3 [\beta] \ldots$ we thus have $K_j \supseteq K$ for all $j$, due to the monotonicity of $[\beta]$. For any $p \in K$, $j \in \mathbb{N}$, $\sigma$ we thus have $p\sigma \in \text{REGION}(qX^\beta\sigma, [X^\beta\sigma] \mathcal{E})$. In particular, for $B = B_{qX^\beta}$ (from Lemma $[11]$) we deduce that

$$p\gamma \in \text{REGION}(qX^\beta\gamma, [X^\beta\gamma] \mathcal{E}) \text{ for each } p \in K. \tag{4}$$

Suppose now that the finite paths $p_{0\alpha\sigma} = c_m \xrightarrow{u} c'_{i-d_1} = qX^\gamma$ and $qX \xrightarrow{w} qX^\beta$ do not constitute a witness; hence $qX^\beta \sim qX^\beta^\omega$. We recall that the bisimilarity class $[c'_i]$ is determined by $q_iX_i\alpha\beta$ and the tuple $(p_{i-\ell})_{p \in K}$ (by Prop. $[10]$). By $[11]$ we deduce that the classes $[p\gamma]$ in the tuple are in the finite set $\text{REGION}(qX^\beta\sigma, [X^\beta\sigma] \mathcal{E})$ in $(L_{\mathcal{M}})_{\sim}$.

We note that there are infinitely many $i$ for which the triple $(q, X, \beta)$ induced as above is the same. If none of these $i$ yields a witness then the bisimilarity classes of infinitely many $c'_i$ are determined by the same $q'X'\alpha\beta$ and the same tuple of bisimilarity classes from $\text{REGION}(qX^\beta\sigma, [X^\beta\sigma] \mathcal{E})$. This implies $c'_i \sim c'_j$ for some $i \neq j$, a contradiction. 

\[ \square \]
Additional remarks

The presented decidability result also holds when we allow a restricted use of \( \epsilon \)-transitions; roughly speaking, some top-pairs \( pX \) can be unstable, in which case any configuration \( pX \alpha \) always silently changes to \( p' \alpha \) where \( p' \) is determined by \( pX \). In fact, Sénizergues [5] showed the decidability of equivalence in this more general setting (and this is handled in the generalization of [3] in the framework of terms as well). On the other hand, if we allow the option that \( pX \) can either silently change to \( p' \) or perform an action due to a rule \( pX \xrightarrow{\alpha} q\alpha \), we get the undecidability of equivalence (as shown in [4]). But this does not answer the decidability question for regularity in this more general setting, neither in the setting of \( \epsilon \)-free second-order pda [2].

The aim of this article has been to present a proof of the decidability of pda regularity w.r.t. bisimilarity. Many questions remain for exploring. E.g., can we avoid the use of the “black-box” procedure deciding equivalence? Can we derive something reasonable in the cases where the equivalence is undecidable (e.g. for general \( \epsilon \)-rules or for \( \epsilon \)-free higher-order pda)? Can we strengthen the known complexity bounds?

The reference list and the discussion of the related work should be surely enhanced; the author will welcome any comments also regarding this.

Author’s acknowledgements. I would like to thank Stefan Göller for drawing my attention to the decidability question for regularity of pushdown processes, and for his information about some previous related works (like [8]).

References

[1] M. Benedikt, S. Göller, S. Kiefer, and A. Murawski. Bisimilarity of pushdown systems is nonelementary. In Proc. LICS 2013. IEEE Computer Society, 2013. To appear (see also http://arxiv.org/abs/1210.7686).
[2] C. H. Broadbent and S. Göller. On bisimilarity of higher-order pushdown automata: Undecidability at order two. In FSTTCS 2012, volume 18 of LIPIcs, pages 160–172. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.
[3] P. Jančar. Decidability of DPDA language equivalence via first-order grammars. In Proc. LICS 2012. IEEE Computer Society, 2012.
[4] P. Jančar and J. Srba. Undecidability of bisimilarity by Defender’s forcing. J. ACM, 55(1), 2008.
[5] G. Sénizergues. The bisimulation problem for equational graphs of finite out-degree. SIAM J. Comput., 34(5):1025–1106, 2005.
[6] J. Srba. Roadmap of infinite results. In Current Trends In Theoretical Computer Science, The Challenge of the New Century, volume 2, pages 337–350. World Scientific Publishing Co., 2004. For an updated version see http://users.cs.au.dk/srba/roadmap/.
[7] C. Stirling. Deciding DPDA equivalence is primitive recursive. In Proc. ICALP’02, volume 2380 of LNCS, pages 821–832. Springer-Verlag, 2002.
[8] L. G. Valiant. Regularity and related problems for deterministic pushdown automata. J. ACM, 22(1):1–10, 1975.