Deciding Semantic Finiteness of Pushdown Processes and First-Order Grammars w.r.t. Bisimulation Equivalence

Petr Jančar

Dept. of Computer Science, FEI, Technical University Ostrava, Czech Republic
petr.jancar@vsb.cz

Abstract

The problem if a given configuration of a pushdown automaton (PDA) is bisimilar with some (unspecified) finite-state process is shown to be decidable. The decidability is proven in the framework of first-order grammars, which are given by finite sets of labelled rules that rewrite roots of first-order terms. The framework is equivalent to PDA where also deterministic popping epsilon-steps are allowed, i.e. to the model for which Sénizergues showed an involved procedure deciding bisimilarity (FOCS 1998). Such a procedure is here used as a black-box part of the algorithm. For deterministic PDA the regularity problem was shown decidable by Valiant (JACM 1975) but the decidability question for nondeterministic PDA, answered positively here, had been open (as indicated, e.g., by Broadbent and Goeller, FSTTCS 2012).

1998 ACM Subject Classification F.4.2 Grammars and Other Rewriting Systems

Keywords and phrases pushdown processes, first-order grammars, bisimulation, regularity

Digital Object Identifier 10.4230/LIPIcs.MFCS.2016.52

1 Introduction

The question of deciding semantic equivalences of systems, like language equivalence, has been a frequent topic in computer science. A closely related question asks if a given system in a class $\mathcal{C}_1$ has an equivalent in a simpler class $\mathcal{C}_2$. Pushdown automata (PDA) constitute a well-known example. Language equivalence and regularity are undecidable for PDA. In the case of deterministic PDA (DPDA), the decidability and complexity results for regularity (see [13] and the references therein) preceded the famous decidability result for equivalence by Sénizergues [9].

In concurrency theory, logic, verification, and other areas, a finer equivalence, called *bisimulation equivalence* or *bisimilarity*, has emerged as another fundamental behavioural equivalence; on deterministic systems it essentially coincides with language equivalence. An on-line survey of the results which study this equivalence in a specific area of process rewrite systems is maintained by Srba [11].

Among the most involved results in this area is the decidability of bisimilarity for pushdown processes, generated by (nondeterministic) PDA with only deterministic and popping ε-steps; this was shown by Sénizergues [10] who thus generalized his above mentioned result for DPDA. There is no known upper bound on the complexity of this decidable problem. The nonelementary lower bound established in [1] is, in fact, TOWER-hardness in the terminology of [8], and it holds even for real-time PDA, i.e. PDA with no ε-steps. For the above mentioned PDA with deterministic and popping ε-steps the bisimilarity problem is even not primitive recursive, its Ackermann-hardness is shown in [5]. In the deterministic case, the equivalence

41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016).
Editors: Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier; Article No. 52; pp. 52:1–52:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

problem is known to be PTIME-hard, and has a primitive recursive upper bound shown by Stirling [12] (where a finer analysis places the problem in TOWER [5]).

Extrapolating the deterministic case, we might expect that for PDA the "regularity" problem w.r.t. bisimilarity (asking if a given PDA-configuration is bisimilar with a state in a finite-state system) is decidable as well, and that this problem might be easier than the equivalence problem solved in [10]; "only" EXPTIME-hardness is known here (see [7], and [11] for detailed references). Nevertheless, this decidability question has been open so far, as also indicated in [2] (besides [11]).

Contribution of this paper. We show that semantic finiteness of pushdown configurations w.r.t. bisimilarity is decidable. The decidability is proven in the framework of *first-order grammars*, i.e. of finite sets of labelled rules that rewrite roots of first-order terms. The framework is equivalent to PDA where also deterministic and popping ε-steps are allowed, i.e. to the model to which Sénizergues's general decidability proof [10] applies. (A simplified proof directly in the first-order grammar framework is given in [4].) The presented algorithm, answering if a given configuration, i.e. a first-order term in the labelled transition system generated by a first-order grammar, has a bisimilar finite-state system, uses the result of [10] (or of [4]) as a black-box procedure. By [5] we cannot get a primitive recursive upper bound via a black-box use of the decision procedure for bisimilarity.

Semidecidability of the semantic finiteness problem has been long clear, hence it is the existence of finite effectively verifiable witnesses of the negative case that is the crucial point here. Such witnesses are shown by considering "limits" of repeated substitutions, resulting in regular terms (i.e.infinite terms with only finitely many subterms). Some finite paths with "pumpable" segments are shown to be increasing the "equivalence-level" with the respective limit above any bound while never reaching the equivalence class of the limit. The (black-box) procedure deciding equivalence is used to show a verifiable bound on the number of segment-pumpings that allows to confirm the witness property of a path.

A full version of this paper is planned to appear as the second version of the paper at `http://arxiv.org/abs/1305.0516`; it will contain detailed proofs.

2 Basic Notions and Result

In this section we define the basic notions and state the result in the form of a theorem. Some standard definitions are restricted when we do not need the full generality. We finish the section by a note about a transformation of pushdown automata to first-order grammars.

By $\mathbb{N}$ and $\mathbb{N}_+$ we denote the sets of nonnegative integers and of positive integers, respectively. By $[i, j]$ we denote the set $\{i, i+1, \dots, j\}$. For a set $\mathcal{A}$, by $\mathcal{A}^*$ we denote the set of finite sequences of elements of $\mathcal{A}$, which are also called *words* (over $\mathcal{A}$). By $|w|$ we denote the *length* of $w \in \mathcal{A}^*$, and by ε the *empty sequence* (hence $|\varepsilon| = 0$). We put $\mathcal{A}^+ = \mathcal{A}^* \smallsetminus \{\varepsilon\}$.

Labelled transition systems. A *labelled transition system*, an *LTS* for short, is a tuple $\mathcal{L} = (\mathcal{S}, \Sigma, (\xrightarrow{a})_{a\in\Sigma})$ where $\mathcal{S}$ is a finite or countable set of *states*, Σ is a finite set of *actions* (or *letters*), and $\xrightarrow{a} \subseteq \mathcal{S} \times \mathcal{S}$ is a set of *a-transitions* (for each $a \in \Sigma$). We say that $\mathcal{L}$ is a *deterministic LTS* if for each pair $s \in \mathcal{S}$, $a \in \Sigma$ there is at most one s' such that $s \xrightarrow{a} s'$ (which stands for $(s, s') \in \xrightarrow{a}$). By $s \xrightarrow{w} s'$, where $w = a_1 a_2 \dots a_n \in \Sigma^*$, we denote that there is a *path* $s = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \cdots \xrightarrow{a_n} s_n = s'$; if $s \xrightarrow{w} s'$, then s' is *reachable from* s. By $s \xrightarrow{w}$ we denote that w is *enabled in* s, i.e., $s \xrightarrow{w} s'$ for some s'. If $\mathcal{L}$ is deterministic, then $s \xrightarrow{w} s'$ or $s \xrightarrow{w}$ denotes a unique path.

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

Semantic finiteness. Given $\mathcal{L} = (\mathcal{S}, \Sigma, (\xrightarrow{a})_{a\in\Sigma})$, we say that $s_0 \in \mathcal{S}$ is *finite up to bisimilarity*, or *bisim-finite* for short, if there is some state f in some finite LTS such that $s_0 \sim f$; otherwise s_0 is *infinite up to bisimilarity*, or *bisim-infinite*. (When comparing states from different LTSs, we implicitly refer to the disjoint union of these LTSs.)

First-order terms, regular terms, finite graph presentations. We will consider LTSs in which states are first-order regular terms. They are built from *variables* from a fixed countable set $\textsc{Var} = \{x_1, x_2, x_3, \dots\}$ and from *function symbols*, also called *(ranked) nonterminals*, from some specified finite set $\mathcal{N}$; each $A \in \mathcal{N}$ has $arity(A) \in \mathbb{N}$. (An example of a finite term is $C(D(x_3, B), x_2)$, where the arities of B, C, D are $0, 2, 2$, respectively.)

Transitions will be determined by a finite set of (schematic) *root-rewriting* rules (that can be exemplified by $A(x_1, x_2, x_3) \xrightarrow{b} C(D(x_3, B), x_2)$, where x_1, x_2, x_3 serve as the "place-holders" for the depth-1 subterms of a term with the root A that might be rewritten by performing action b). We will now formalize this, making also some conventions on the use of (finite and infinite) terms and substitutions.

We identify terms with their syntactic trees. Thus a *term over* $\mathcal{N}$ is (viewed as) a rooted, ordered, finite or infinite tree where each node has a label from $\mathcal{N} \cup \textsc{Var}$; if the label of a node is $x_i \in \textsc{Var}$, then the node has no successors, and if the label is $A \in \mathcal{N}$, then it has m (immediate) successor-nodes where $m = arity(A)$. A subtree of a term E is also called a *subterm* of E. We make no difference between isomorphic (sub)trees, and thus a subterm can have more (maybe infinitely many) *occurrences* in E. Each *subterm-occurrence* has its (nesting) *depth in* E, which is its (naturally defined) distance from the root of E. We also use the standard notation for terms: we write $E = x_i$ or $E = A(G_1, \dots, G_m)$ with the obvious meaning; in the latter case we have $\textsc{root}(E) = A \in \mathcal{N}$, $m = arity(A)$, and $G_1, \dots, G_m$ are the ordered occurrences of depth-1 subterms of E.

A *term* is *finite* if the respective tree is finite. A (possibly infinite) *term* is *regular* if it has only finitely many subterms (though the subterms may be infinite and can have infinitely many occurrences). We note that any regular term has at least one *finite-graph presentation*, i.e. a finite directed graph, with a designated root, where each node has a label from $\mathcal{N} \cup \textsc{Var}$; if the label of a node is $x_i \in \textsc{Var}$, then the node has no outgoing arcs, if the label is $A \in \mathcal{N}$, then it has m ordered outgoing arcs where $m = arity(A)$. The standard tree-unfolding of the graph is the respective term, which is infinite if there are cycles in the graph. The nodes in the least presentation of E are bijectively mapped onto (the roots of) the subterms of E.

In what follows, by a "term" we mean a "regular term" unless the context makes clear that the term is finite. (We do not consider non-regular terms.) We reserve symbols A, B, C, D to range over nonterminals, and E, F, G, H to range over (regular) terms.

Substitutions, associative composition, limits of infinite compositions. By $\textsc{Terms}_{\mathcal{N}}$ we denote the set of all (regular) terms over a set $\mathcal{N}$ of (ranked) nonterminals (and over the set $\textsc{Var}$ of variables). A *substitution* σ is a mapping $\sigma : \textsc{Var} \to \textsc{Terms}_{\mathcal{N}}$ whose *support* $\textsc{supp}(\sigma) = \{x_i \mid \sigma(x_i) \neq x_i\}$ is *finite*; we reserve the symbol σ for substitutions. By *applying* *a substitution* σ to a term E we get the term $E\sigma$ that arises from E by replacing each occurrence of x_i with $\sigma(x_i)$; given graph presentations, in the graph of E we just redirect each arc leading to x_i towards the root of $\sigma(x_i)$ (which includes the special "root-designating arc" when $E = x_i$). Hence $E = x_i$ implies $E\sigma = x_i\,\sigma = \sigma(x_i)$.

The natural *composition of substitutions*, where $\sigma = \sigma_1\sigma_2$ is defined by $x_i\sigma = (x_i\sigma_1)\sigma_2$, can be easily verified to be associative. We thus write simply $E\sigma_1\sigma_2$ when meaning $(E\sigma_1)\sigma_2$ or $E(\sigma_1\sigma_2)$. We let σ^0 be the empty-support substitution, and we put $\sigma^{i+1} = \sigma\sigma^i$. If σ is *guarded*, which means that $x_i\sigma = x_j$ implies $i = j$ (in other words, for each $x_i \in \textsc{supp}(\sigma)$ the root of $E_i = x_i\sigma$ is a nonterminal "guarding" the occurrences of variables in E_i), then even the limit σ^ω is well-defined: "operationally", to get graph presentations of terms $x_i\sigma^\omega$ from graph presentations of $x_i\sigma$, for all $x_i \in \textsc{supp}(\sigma)$, we redirect any arc leading to x_j, where $x_j \in \textsc{supp}(\sigma)$, towards the root of (the presentation of) $x_j\sigma$. We note that no variable $x_i \in \textsc{supp}(\sigma)$ occurs in any term $E\sigma^\omega$, for any guarded substitution σ; such variables "disappear" by applying σ^ω. (Hence $E\sigma^\omega$ can only contain variables x_i for which $x_i\sigma = x_i$.)

First-order grammars. A *first-order grammar*, or just a *grammar* for short, is a tuple $\mathcal{G} = (\mathcal{N}, \Sigma, \mathcal{R})$ where $\mathcal{N} = \{A_1, A_2, \dots\}$ is a finite set of ranked *nonterminals*, viewed as function symbols with arities, $\Sigma = \{a_1, a_2, \dots\}$ is a finite set of *actions* (or letters), and $\mathcal{R} = \{r_1, r_2, \dots\}$ is a finite set of *rules* of the form

$$A(x_1, x_2, \dots, x_m) \xrightarrow{a} E \tag{1}$$

where $A \in \mathcal{N}$, $arity(A) = m$, $a \in \Sigma$, and E is a *finite* term over $\mathcal{N}$ in which each occurring variable is from the set $\{x_1, x_2, \dots, x_m\}$. We can exemplify the rules by $A(x_1, x_2, x_3) \xrightarrow{b} C(D(x_3, B), x_2)$, $A(x_1, x_2, x_3) \xrightarrow{b} x_2$, $D(x_1, x_2) \xrightarrow{a} A(D(x_2, x_2), x_1, B)$; here the arities of A, B, C, D are $3, 0, 2, 2$, respectively.

A rule $A(x_1, x_2, \dots, x_m) \xrightarrow{a} E$ will generate a-transitions $A(x_1, x_2, \dots, x_m)\sigma \xrightarrow{a} E\sigma$ for all substitutions σ. The concrete rule $A(x_1, x_2, x_3) \xrightarrow{b} C(D(x_3, B), x_2)$ generates the transitions like $A(x_1, x_2, x_3) \xrightarrow{b} C(D(x_3, B), x_2)$ and $A(x_5, x_5, x_2) \xrightarrow{b} C(D(x_2, B), x_5)$, and more generally $A(G_1, G_2, G_3) \xrightarrow{b} C(D(G_3, B), G_2)$ for any (regular) terms G_1, G_2, G_3. The rule $A(x_1, x_2, x_3) \xrightarrow{b} x_2$ generates $A(G_1, G_2, G_3) \xrightarrow{b} G_2$. We now give a more formal definition.

LTSs generated by grammars. Given $\mathcal{G} = (\mathcal{N}, \Sigma, \mathcal{R})$, by $\mathcal{L}^{\text{R}}_{\mathcal{G}}$ we denote the (*rule-based*) LTS $\mathcal{L}^{\text{R}}_{\mathcal{G}} = (\textsc{Terms}_{\mathcal{N}}, \mathcal{R}, (\xrightarrow{r})_{r \in \mathcal{R}})$ where each rule r of the form $A(x_1, x_2, \dots, x_m) \xrightarrow{a} E$ induces transitions $A(x_1, \dots, x_m)\sigma \xrightarrow{r} E\sigma$ for any substitution σ (also unguarded; we can have $x_i\sigma = x_j$ for $i \neq j$). Thus the rule $A(x_1, \dots, x_m) \xrightarrow{r} E$ is itself a transition, using σ with $\textsc{supp}(\sigma) = \emptyset$.
The LTS $\mathcal{L}^{\text{R}}_{\mathcal{G}}$ is deterministic, since for each F and r there is at most one H such that $F \xrightarrow{r} H$. We note that *variables* are *dead* (have no outgoing transitions), and *transitions cannot add variables*, i.e., $F \xrightarrow{w} H$ implies that each variable occurring in H also occurs in F (but not necessarily vice versa).

Since the rhs (right-hand sides) E in the rules (1) are finite, all terms reachable from a finite term are finite. (It is convenient to have the rhs finite while including regular terms into our LTSs; the other options are in principle equivalent.)

The deterministic rule-based LTS $\mathcal{L}^{\text{R}}_{\mathcal{G}}$ is helpful technically, but we are primarily interested in the (generally nondeterministic) *action-based* LTS $\mathcal{L}^{\text{A}}_{\mathcal{G}} = (\textsc{Terms}_{\mathcal{N}}, \Sigma, (\xrightarrow{a})_{a \in \Sigma})$ where each rule $A(x_1,\dots,x_m) \xrightarrow{a} E$ induces the transitions $A(x_1,\dots,x_m)\sigma \xrightarrow{a} E\sigma$ for all substitutions σ.

Given a grammar $\mathcal{G} = (\mathcal{N}, \Sigma, \mathcal{R})$, two *terms* from $\textsc{Terms}_{\mathcal{N}}$ are *bisimilar* if they are bisimilar as states in the action-based LTS $\mathcal{L}^{\mathrm{A}}_{\mathcal{G}}$. By our definitions all variables are bisimilar, since they are dead terms. The variables serve us primarily as "place-holders for subterm-occurrences" in terms (that might themselves be variable-free); such a use of variables has been already exemplified in the rules (1).

Main result, and its relation to pushdown automata. We now state the theorem, to be proven in the next section, and we mention why the result also applies to pushdown automata (PDA) with deterministic popping ε-steps.

▶ **Theorem 1.** *There is an algorithm that, given a grammar $\mathcal{G} = (\mathcal{N}, \Sigma, \mathcal{R})$ and (a finite presentation of) $E_0 \in \textsc{Terms}(\mathcal{N})$, decides if E_0 is bisim-finite (i.e., if $E_0 \sim f$ for a state f in some finite LTS).*

A transformation of (nondeterministic) PDA in which deterministic popping ε-steps are allowed to first-order grammars (with no ε-steps) is recalled in the full arxiv-version. This makes clear that the semantic finiteness of PDA with deterministic popping ε-steps (w.r.t. bisimilarity) is also decidable. In fact, the problems are interreducible; the close relationship between (D)PDA and first-order schemes has been long known (see, e.g., [3]). The proof of Theorem 1 presented here uses the fact that bisimilarity of first-order grammars is decidable; this was shown for the above mentioned PDA model by Sénizergues [10], and a direct proof in the first-order-term framework was presented in [4]. We note that for PDA where popping ε-steps can be in conflict with "visible" steps bisimilarity is already undecidable [6]; hence the proof presented here does not yield the decidability of semantic finiteness in this more general model.

3 Proof of Theorem 1

3.1 Computability of eq-levels, and semidecidability of bisim-finiteness

We will note that the semidecidability of bisim-finiteness is clear, but we first recall the computability of eq-levels, which is one crucial ingredient in our proof of semidecidability of bisim-infiniteness.

Stratified equivalence, and eq-levels. Assuming an LTS $\mathcal{L} = (\mathcal{S}, \Sigma, (\xrightarrow{a})_{a\in\Sigma})$, we put $\sim_0 = \mathcal{S}\times\mathcal{S}$, and define $\sim_{k+1} \subseteq \mathcal{S}\times\mathcal{S}$ (for $k\in\mathbb{N}$) as the set of pairs covered by $\sim_k$. (Hence $s \sim_{k+1} t$ iff for any $s \xrightarrow{a} s'$ there is $t \xrightarrow{a} t'$ such that $s' \sim_k t'$ and for any $t \xrightarrow{a} t'$ there is $s \xrightarrow{a} s'$ such that $s' \sim_k t'$.)

We easily verify that $\sim_k$ are equivalence relations, and that $\sim_0 \supseteq \sim_1 \supseteq \sim_2 \supseteq \cdots\cdots \supseteq \sim$. For the (first infinite) ordinal ω we put $s \sim_\omega t$ if $s \sim_k t$ for all $k\in\mathbb{N}$; hence $\sim_\omega = \bigcap_{k\in\mathbb{N}} \sim_k$. It is standard (and can be easily checked) that $\bigcap_{k\in\mathbb{N}} \sim_k$ is a bisimulation in image-finite LTSs, and thus $\sim = \bigcap_{k\in\mathbb{N}} \sim_k = \sim_\omega$. We recall that $\mathcal{L}$ is *image-finite* if the set $\{s' \mid s \xrightarrow{a} s'\}$ is finite for each pair $s\in\mathcal{S}$, $a\in\Sigma$. Our grammar-generated LTSs $\mathcal{L}^{\mathrm{A}}_{\mathcal{G}}$ are obviously image-finite (while $\mathcal{L}^{\mathrm{R}}_{\mathcal{G}}$ are even deterministic); we thus further assume image-finiteness.

We attach the *equivalence level* (eq-level) $\textsc{EqLv}(s,t) = \max\{k \in \mathbb{N}\cup\{\omega\} \mid s \sim_k t\}$ to each pair of states.

Eq-levels are computable for first-order grammars. We now state an important lemma that follows easily from the involved decidability proof in [10] (and a transformation to first-order grammars); as already mentioned, a proof given directly for the first-order grammars was presented in [4]. (This is surely a *fundamental theorem* in general, the name *lemma* has been chosen here to reflect that it is a prerequisite for the only theorem proven in this paper.)

▶ **Lemma 2.** *There is an algorithm that, given $\mathcal{G} = (\mathcal{N}, \Sigma, \mathcal{R})$ and $E_0, F_0 \in \textsc{Terms}(\mathcal{N})$, computes $\textsc{EqLv}(E_0, F_0)$ in $\mathcal{L}^{\mathsf{A}}_{\mathcal{G}}$ (and thus also decides if $E_0 \sim F_0$).*

Proof. For each fixed $k \in \mathbb{N}$ it is decidable if $E_0 \sim_k F_0$, as can be shown by a straightforward induction on k. The question $E_0 \overset{?}{\sim} F_0$, i.e. $E_0 \overset{?}{\sim}_\omega F_0$, can be decided by [10] (and [4]). ◀

Semidecidability of bisim-finiteness. Given $\mathcal{G}$ and E_0, we can systematically generate all finite LTSs, presenting them by first-order grammars with nullary nonterminals (which then coincide with states); for each state f of each generated system we can check if $E_0 \sim f$ by Lemma 2. In fact, Lemma 2 is not crucial here, since decidability of $E_0 \sim f$ can be shown in a much simpler way (see, e.g., [7]).

3.2 Semidecidability of bisim-infiniteness.

In Section 3.2.1 we note a few simple general facts on bisim-infiniteness, and also note the obvious compositionality (congruence properties) of bisimulation equivalence in our framework of first-order terms. In Section 3.2.2 we describe some finite structures that are candidates for witnessing bisim-infiniteness of a given term, and show an algorithm checking if a candidate is indeed a witness. In Section 3.2.3 we then show that each bisim-infinite term has a witness. Together this yields a proof of Theorem 1.

3.2.1 Some facts on bisim-infiniteness, and compositionality

Bisimilarity quotient. Given an LTS $\mathcal{L} = (\mathcal{S}, \Sigma, (\xrightarrow{a})_{a\in\Sigma})$, the *quotient-LTS* $\mathcal{L}_\sim$ is the tuple $(\{\, [s]; s \in \mathcal{S}\,\}, \Sigma, (\xrightarrow{a})_{a\in\Sigma})$ 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, $[s] \xrightarrow{a} [t]$ implies that for each $s' \in [s]$ there is $t' \in [t]$ such that $s' \xrightarrow{a} t'$. We have $s \sim [s]$, since $\{(s, [s]) \mid s \in \mathcal{S}\}$ is a bisimulation (in the union of $\mathcal{L}$ and $\mathcal{L}_\sim$). We refer to the states of $\mathcal{L}_\sim$ as to the *bisim-classes* (of $\mathcal{L}$).

A sufficient condition for bisim-infiniteness. We recall that $s_0 \in \mathcal{S}$ is *bisim-finite* if there is some state f in a finite LTS such that $s_0 \sim f$; otherwise s_0 is *bisim-infinite*. We observe that s_0 is bisim-infinite in $\mathcal{L}$ iff the reachability set of $[s_0]$ in $\mathcal{L}_\sim$, i.e. the set of states reachable from $[s_0]$ in $\mathcal{L}_\sim$, is infinite. The LTSs generated by first-order grammars are *finitely branching* (i.e., the set $\{s' \mid s \xrightarrow{a} s' \text{ for some } a\}$ is finite for each $s \in \mathcal{S}$), and we also use (one implication in) the following simple fact:

▶ **Proposition 3.** *A state s_0 of a finitely branching LTS is bisim-infinite iff there is an infinite path $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} \cdots$ where $s_i \not\sim s_j$ for all $i \neq j$.*

To demonstrate that s_0 is bisim-infinite, it suffices to show that its reachability set contains states with arbitrarily large *finite* eq-levels w.r.t. a "test state" t. The sufficiency of this condition is based on the simple fact that $s \sim s'$ implies $\textsc{EqLv}(s, t) = \textsc{EqLv}(s', t)$. More formally:

▶ **Proposition 4.** *Given $\mathcal{L} = (\mathcal{S}, \Sigma, (\xrightarrow{a})_{a\in\Sigma})$ and states s_0, t, if for every $e \in \mathbb{N}$ there is s' that is reachable from s_0 and satisfies $e < \textsc{EqLv}(s', t) < \omega$, then s_0 is bisim-infinite.*

Eq-levels w.r.t. a test set in a bounded region. Our final general observation (tailored to a later use) is also straightforward: if two states are bisimilar, then the states in their equally bounded reachability regions must yield the same eq-levels when compared with states from a fixed (test) set. We formalize this observation as follows.

For any $s \in \mathcal{S}$ and $d \in \mathbb{N}$ (a distance, or a "radius") we put

$$\textsc{Region}(s,d) = \{s' \mid s \xrightarrow{w} s' \text{ for some } w \in \Sigma^* \text{ where } |w| \leq d\}.$$

For any $s \in \mathcal{S}$, $d \in \mathbb{N}$, and $\mathcal{T} \subseteq \mathcal{S}$ (a test set), we define the following subset of $\mathbb{N}$ (*finite* TestEqLevels):

$$\mathrm{TEL}(s,d,\mathcal{T}) = \{e \in \mathbb{N} \mid e = \textsc{EqLv}(s',t) \text{ for some } s' \in \textsc{Region}(s,d) \text{ and some } t \in \mathcal{T}\}.$$

▶ **Proposition 5.** *If* $\mathrm{TEL}(s,d,\mathcal{T}) \neq \mathrm{TEL}(s',d,\mathcal{T})$ *then* $s \not\sim s'$*.*

Compositionality of the states of the grammar-generated LTSs. Regarding the congruence properties, in principle it suffices for us to observe that if in a term E we replace a subterm F with F' such that $F' \sim F$ then the resulting term E' satisfies $E' \sim E$ (replacing a subterm with an equivalent one does not change the bisim-class). Hence $A(G_1,\dots,G_m) \not\sim A(G'_1,\dots,G'_m)$ implies that $G_i \not\sim G'_i$ for some $i \in [1,m]$. (This is surely not specific to bisimilarity.) Formally, we put $\sigma \sim \sigma'$ if $x_i\sigma \sim x_i\sigma'$ for each x_i, and we note:

▶ **Proposition 6.** *If* $\sigma \sim \sigma'$*, then* $E\sigma \sim E\sigma'$*.*
(Hence $E\sigma \not\sim E\sigma'$ *implies that* $x_i\sigma \not\sim x_i\sigma'$ *for some* x_i *occurring in* E*.)*

Conventions. We further consider only the *normalized* grammars $\mathcal{G} = (\mathcal{N},\Sigma,\mathcal{R})$, i.e. those satisfying the following condition: for any $A(x_1,\dots,x_m)$ and any $i \in [1,m]$ there is a word w such that $A(x_1,\dots,x_m) \xrightarrow{w} x_i$; hence for any E it is possible to "sink" to any of its subterm-occurrences by applying the grammar-rules. Such a normalization can be efficiently achieved by harmless modifications of the nonterminal arities and of the rules in $\mathcal{R}$, while the LTS $\mathcal{L}^{\mathrm{A}}_{\mathcal{G}}$ remains the same up to isomorphism.

For convenience, in our notation we use m as the arity of all nonterminals in the considered grammar, though formally the *maximum* arity is meant. We will thus harmlessly write $A(G_1,\dots,G_m)$ instead of $A(G_1,\dots,G_{m_A})$ where $m_A = \mathit{arity}(A)$. (In fact, such uniformity can be achieved while keeping the above normalization condition, when a slight problem with arity 0 is handled; but this is not necessary for us to discuss.)

From now on, we view the expressions like $G \xrightarrow{w} H$ as referring to the deterministic LTS $\mathcal{L}^{\mathrm{R}}_{\mathcal{G}}$ (hence $w \in \mathcal{R}^*$), though $\sim_k$, $\sim$, and the eq-levels refer solely to (the action-based LTS) $\mathcal{L}^{\mathrm{A}}_{\mathcal{G}}$.

3.2.2 Simple witnesses of bisim-infiniteness

We fix a grammar $\mathcal{G} = (\mathcal{N},\Sigma,\mathcal{R})$. Before defining the candidates for witnesses of bisim-infiniteness, we discuss some building segments of ("non-sinking") paths in the LTS $\mathcal{L}^{\mathrm{R}}_{\mathcal{G}}$.

Stairs, direct stairs, simple stairs, stairs eligible for "pumping". A nonempty sequence of rules $w = r_1r_2\dots r_\ell \in \mathcal{R}^+$ is a *stair* if we have $A(x_1,\dots,x_m) \xrightarrow{w} F$ where the rule r_1 is of the form $A(x_1,\dots,x_m) \xrightarrow{a} E$, and F has a nonterminal-root (hence F is not a variable x_i, i.e., the path $A(x_1,\dots,x_m) \xrightarrow{w} F$ does not "sink"); such w is a *direct stair* if there is no v such that $|v| < |w|$ and $A(x_1,\dots,x_m) \xrightarrow{v} F$. If (the above) w is a direct stair and F is a subterm of E (the right-hand side of r_1), then w is a *simple stair.* A *stair* w has the *type* (A,B) ("from A to B") if $A(x_1,\ldots,x_m) \xrightarrow{w} F$ where $\text{ROOT}(F) = B$.

(E.g., the sequence r_1r_2 of rules used in the path $A(G_1,G_2,G_3) \xrightarrow{r_1} C(D(G_3,B),G_2) \xrightarrow{r_2} D(G_3,B)$ is a stair, of type (A,D), that might be a simple stair; on the other hand r_2 is no stair. Some simple stairs can be of the form $A(x_1,\ldots,x_m) \xrightarrow{w} A'(x_{i_1},\ldots,x_{i_m})$, where $\{i_1,\ldots,i_m\} \subseteq \{1,\ldots,m\}$; we might even have $A = A'$ but in this case $(x_1,\ldots,x_m) \neq (x_{i_1},\ldots,x_{i_m})$ since $A(x_1,\ldots,x_m) \xrightarrow{w} A(x_1,\ldots,x_m)$ is no direct stair.)

It is easy to observe that any direct stair w is a sequence of compatible simple stairs, i.e., $w = w_1w_2\ldots w_n$ where w_i is a simple stair of type (A_{i-1},A_i), for each $i \in [1,n]$; we thus have $A_0(x_1,\ldots,x_m) \xrightarrow{w_1} A_1(x_1,\ldots,x_m)\sigma_1 \xrightarrow{w_2} A_2(x_1,\ldots,x_m)\sigma_2\sigma_1 \xrightarrow{w_3} \cdots$ for the respective (not necessarily guarded) substitutions σ_i.

A *stair* w where $A(x_1,\ldots,x_m) \xrightarrow{w} A(E_1,\ldots,E_m)$ is *eligible* (for "pumping") if the set of "root-sticks" $R = \{x_i \mid E_j = x_i \text{ for some } j \in [1,m]\}$ is equal to $\{x_i \mid i \in [1,m], E_i = x_i\}$.

(E.g., the stair $A(x_1,x_2,x_3,x_4) \xrightarrow{w} A(x_1,B(x_2,x_2,x_4,x_1),x_3,x_3)$ is eligible, with $R = \{x_1,x_3\}$. The stair $A(x_1,x_2,x_3,x_4) \xrightarrow{v} A(x_2,B(x_2,x_2,x_4,x_1),x_3,x_3)$ is not eligible but the respective "double" stair $A(x_1,x_2,x_3,x_4) \xrightarrow{vv} A(B(x_2,x_2,x_4,x_1),B(\ldots),x_3,x_3)$ is eligible. In particular, if in $A(x_1,\ldots,x_m) \xrightarrow{w} A(E_1,\ldots,E_m)$ all E_j have nonterminal-roots, then w is eligible, with $R = \emptyset$.)

An important fact is that for any eligible stair w, where $A(x_1,\ldots,x_m) \xrightarrow{w} A(x_1,\ldots,x_m)\sigma$, we can define the terms $G_{(w,z)}$ for all $z \in \mathbb{N} \cup \{\omega\}$ by putting

$$A(x_1,\ldots,x_m) \xrightarrow{w^z} A(x_1,\ldots,x_m)\sigma^z = G_{(w,z)}$$

which is well defined also for $z = \omega$. (Though σ might be not guarded, we have that $x_j\sigma = x_i$ for $i \neq j$ implies $x_i\sigma = x_i$ due to the eligibility and thus $x_j\sigma^\omega = x_i$.)

Candidates for simple witnesses of bisim-infiniteness. Given a grammar $\mathcal{G} = (\mathcal{N},\Sigma,\mathcal{R})$ and a term E_0, by a *candidate for a simple witness* (of bisim-infiniteness of E_0), or by a *candidate* for short, we mean a pair (u,w) where $u \in \mathcal{R}^*$, $w \in \mathcal{R}^+$, $E_0 \xrightarrow{uw}$, and w is an eligible stair, of the form $A(x_1,\ldots,x_m) \xrightarrow{w} A(x_1,\ldots,x_m)\sigma$; we thus have

$$E_0 \xrightarrow{u} A(x_1,\ldots,x_m)\sigma_0 \xrightarrow{w} G_{(w,1)}\sigma_0 \xrightarrow{w} G_{(w,2)}\sigma_0 \xrightarrow{w} G_{(w,3)}\sigma_0 \xrightarrow{w} \cdots$$

for the respective substitution σ_0. We have $G_{(w,j)} = A(x_1,\ldots,x_m)\sigma^j$, and we denote the term $G_{(w,\omega)} = A(x_1,\ldots,x_m)\sigma^\omega$ also by LIM.

We now formalize the simple observation that the terms $G_{(w,k)}\sigma_0$ with increasing $k \in \mathbb{N}$ "approach" the term $\text{LIM}\,\sigma_0$ syntactically, and thus also semantically.

Top-tails presentations. Given a term G and (depth) $d \in \mathbb{N}$, let $\text{NOD}_1,\text{NOD}_2,\ldots,\text{NOD}_n$ ($n \geq 0$) be the ordered nodes of (the syntactic tree of) G in depth d (if there are some); let $F_1,F_2,\ldots,F_n$ be the (occurrences of) subterms of G rooted in $\text{NOD}_1,\text{NOD}_2,\ldots,\text{NOD}_n$.

- By $\text{TOP}_d(G)$ we denote the term that coincides with G up to depth $d-1$ while its ordered nodes in depth d are (leaves) labelled with $x_1,x_2,\ldots,x_n$, respectively; here we assume that no x_i, $i \in [1,n]$, occurs in G in the depths less than d.
- By $\text{TAILS}_d(G)$ we mean the substitution defined by $x_i\,\text{TAILS}_d(G) = F_i$ for $i \in [1,n]$.

We have $G = (\text{TOP}_d(G))\sigma$ where $\sigma = \text{TAILS}_d(G)$. In particular, if $G = A(F_1,\ldots,F_m)$, then $G = (\text{TOP}_1(G))\text{TAILS}_1(G) = A(x_1,\ldots,x_m)\sigma$ where $x_i\sigma = F_i$.

If some x_i occur in G in the depths less than d, then we define $\text{TOP}_d(G)$, $\text{TAILS}_d(G)$ by introducing the variables (in the role of place-holders) other than such x_i. In the following example we highlight this by using "another set of variables" y_i.

For $G = A(B(x_3, x_2, x_2), x_2, C(x_1, B(x_2, x_3, x_1), x_1))$ we have

- $\text{TOP}_2(G) = A(B(y_1, y_2, y_3), x_2, C(y_4, y_5, y_6))$, and
- $\text{TAILS}_2(G) = \{(y_1, x_3), (y_2, x_2), (y_3, x_2), (y_4, x_1), (y_5, B(x_2, x_3, x_1)), (y_6, x_1)\}$.

The next proposition refers to a candidate $E_0 \xrightarrow{u} A(x_1, \ldots, x_m)\sigma_0 \xrightarrow{w} A(x_1, \ldots, x_m)\sigma\sigma_0$ where we denote $x_i\sigma$ by E_i for $i \in [1, m]$.

▶ **Proposition 7.** *The following conditions hold for all $k \in \mathbb{N}$ and $i \in [1, m]$.*

1. $\text{TOP}_k(E_i\sigma^k\sigma_0) = \text{TOP}_k(E_i\sigma^\omega\sigma_0)$, *hence* $\text{TOP}_k(G_{(w,k)}\sigma_0) = \text{TOP}_k(\text{LIM}\,\sigma_0)$.
2. $E_i\sigma^k\sigma_0 \sim_k E_i\sigma^\omega\sigma_0$ *and thus* $\text{EQLV}(G_{(w,k)}\sigma_0, \text{LIM}\,\sigma_0) \geq k$.

Checking if a candidate is a simple witness. A candidate $E_0 \xrightarrow{u} A(x_1, \ldots, x_m)\sigma_0 \xrightarrow{w}$ is a *simple witness* (of bisim-infiniteness of E_0) if $G_{(w,k)}\sigma_0 \not\sim \text{LIM}\,\sigma_0$ for infinitely many $k \in \mathbb{N}$. Since $\text{EQLV}(G_{(w,k)}\sigma_0, \text{LIM}\,\sigma_0) \geq k$ (Prop. 7(2)), we then have

$$e < \text{EQLV}(G_{(w,e+1)}\,\sigma_0, \text{LIM}\,\sigma_0) < \omega \text{ for infinitely many } e \in \mathbb{N},$$

and by Prop. 4 we derive that E_0 is bisim-infinite if it has a simple witness.

The existence of an algorithm checking if a candidate is a simple witness follows from the next lemma, if we recall the fundamental fact captured by Lemma 2.

▶ **Lemma 8.** *Given a candidate $E_0 \xrightarrow{u} A(x_1, \ldots, x_m)\sigma_0 \xrightarrow{w} A(E_1, \ldots, E_m)\sigma_0$, there is a computable number e such that one of the following conditions holds:*

1. $G_{(w,e)}\sigma_0 \sim \text{LIM}\,\sigma_0$, *in which case* $G_{(w,k)}\sigma_0 \sim \text{LIM}\,\sigma_0$ *for all* $k \geq e$, *or*
2. $G_{(w,e)}\sigma_0 \not\sim \text{LIM}\,\sigma_0$, *in which case* $G_{(w,k)}\sigma_0 \not\sim \text{LIM}\,\sigma_0$ *for all* $k \geq e$.

(The candidate is a simple witness of bisim-infiniteness of E_0 in the case 2.)

Proof. We restrict our attention to those E_i that "almost always matter" when we apply σ, where $x_i\sigma = E_i$, to the term $A(E_1, \ldots, E_m)\sigma^\ell$, for growing $\ell \in \mathbb{N}$. The sets

$$\mathcal{V}_\ell = \{x_i \mid x_i \text{ occurs in } A(E_1, \ldots, E_m)\sigma^\ell\}$$

satisfy $\mathcal{V}_{\ell+1} = \{x_j \mid x_j \text{ occurs in } E_i \text{ for some } i \text{ such that } x_i \in \mathcal{V}_\ell\}$, and thus $\{x_1, \ldots, x_m\} \supseteq \mathcal{V}_0 \supseteq \mathcal{V}_1 \supseteq \mathcal{V}_2 \supseteq \cdots$. Let ℓ_0 be the smallest such that $\mathcal{V}_{\ell_0} = \mathcal{V}_{\ell_0+1}$ ($= \mathcal{V}_{\ell_0+2} = \cdots$); hence $\ell_0 \leq m$.

We can compute a number d (the "radius" for the below defined region that will be used in the application of Prop. 5) so that within d transition-steps we can reach any variable $x_i \in \mathcal{V}_{\ell_0}$ from both $A(E_1, \ldots, E_m)\sigma^{\ell_0}$ and $A(E_1, \ldots, E_m)\sigma^{\ell_0+1}$.

Suppose now that $A(E_1, \ldots, E_m)\sigma^k\sigma_0 \not\sim A(E_1, \ldots, E_m)\sigma^\omega\sigma_0$ (i.e., $G_{(w,k+1)}\,\sigma_0 \not\sim \text{LIM}\,\sigma_0$) and $k > \ell_0$; then by compositionality (Prop. 6) we deduce that $x_i\sigma^{k-\ell_0}\sigma_0 \not\sim x_i\sigma^\omega\sigma_0$ for some $x_i \in \mathcal{V}_{\ell_0}$, and also $x_{i'}\sigma^{k-\ell_0-1}\sigma_0 \not\sim x_{i'}\sigma^\omega\sigma_0$ for some $x_{i'} \in \mathcal{V}_{\ell_0}$. In other words,

$$E_i\sigma^{k-\ell_0-1}\sigma_0 \not\sim E_i\sigma^\omega\sigma_0, \text{ and } E_{i'}\sigma^{k-\ell_0-2}\sigma_0 \not\sim E_{i'}\sigma^\omega\sigma_0 \text{ for some } i, i' \text{ such that } x_i, x_{i'} \in \mathcal{V}_{\ell_0}.$$

Since x_i occurs in $A(E_1, \ldots, E_m)\sigma^{\ell_0}$ and $x_{i'}$ occurs in $A(E_1, \ldots, E_m)\sigma^{\ell_0+1}$, we have that both $E_i\sigma^{k-\ell_0-1}\sigma_0$ and $E_{i'}\sigma^{k-\ell_0-2}\sigma_0$ are in $\text{REGION}(G_{(w,k+1)}\,\sigma_0, d)$, for the above defined "radius" d.

We would like to deduce that also $G_{(w,k+2)}\,\sigma_0 \not\sim \text{LIM}\,\sigma_0$, using the fact that $E_i\sigma^{k-\ell_0-1}\sigma_0$ is in $\text{REGION}(G_{(w,k+2)}\sigma_0, d)$ (though maybe farther than previously but still within the bound d). But this deduction is unsubstantiated in general.

Hence we recall Prop. 5, and compute the maximum MAX_{TEL} in the set $\text{TEL}(\text{LIM}\,\sigma_0, d, \mathcal{T})$ where the test set is $\mathcal{T} = \{E_i\sigma^\omega\sigma_0 \mid x_i \in \mathcal{V}_{\ell_0}\}$. (We can compute this set by Lemma 2.) Let

$$e = \text{MAX}_{\text{TEL}} + m + 3. \text{ Hence } e \geq \text{MAX}_{\text{TEL}} + \ell_0 + 3.$$

If we return to the above analysis of the case $A(E_1,\ldots,E_m)\sigma^k\sigma_0 \not\sim A(E_1,\ldots,E_m)\sigma^\omega\sigma_0$, now assuming $k \geq e$, then the fact that $E_i\sigma^{k-\ell_0-1}\sigma_0$ (for which $E_i\sigma^{k-\ell_0-1}\sigma_0 \not\sim E_i\sigma^\omega\sigma_0$) is also present in $\textsc{Region}(G_{(w,k+2)}\,\sigma_0, d)$ indeed testifies that $G_{(w,k+2)}\,\sigma_0 \not\sim \textsc{Lim}\,\sigma_0$.

(Since $E_i\sigma^{k-\ell_0-1}\sigma_0 \sim_{k-\ell_0-1} E_i\sigma^\omega\sigma_0$, and $k-\ell_0-1 \geq e-\ell_0-1 > \textsc{Max}_{\text{TEL}}$, the value $\textsc{EqLv}(E_i\sigma^{k-\ell_0-1}\sigma_0, E_i\sigma^\omega\sigma_0)$ is finite but bigger than $\textsc{Max}_{\text{TEL}}$; we thus must have $G_{(w,k+2)}\,\sigma_0 \not\sim \textsc{Lim}\,\sigma_0$ by Prop. 5.)

Then $G_{(w,k+2)}\,\sigma_0 \not\sim \textsc{Lim}\,\sigma_0$ similarly entails that $G_{(w,k+3)}\,\sigma_0 \not\sim \textsc{Lim}\,\sigma_0$, etc.

To finish a demonstration that the above e proves the claim, we note that $G_{(w,e)}\,\sigma_0 \sim \textsc{Lim}\,\sigma_0$ entails that we have $E_i\sigma^{e-\ell_0-1}\sigma_0 \sim E_i\sigma^\omega\sigma_0$ for all i (where $x_i \in \mathcal{V}_{\ell_0}$), since otherwise the sets $\text{TEL}(G_{(w,e)}\sigma_0, d, \mathcal{T})$ and $\text{TEL}(\textsc{Lim}\,\sigma_0, d, \mathcal{T})$ would differ. Then $G_{(w,k)}\sigma_0 \sim \textsc{Lim}\,\sigma_0$ for all $k \geq e$ by compositionality. ◀

3.2.3 Each bisim-infinite term has a simple witness

Once we show the next lemma, the proof of Theorem 1 will be finished.

▶ **Lemma 9.** *For any grammar $\mathcal{G}$ and any bisim-infinite E_0 there is a simple witness satisfying the condition 2 in Lemma 8 ($G_{(w,e)}\sigma_0 \not\sim \textsc{Lim}\,\sigma_0$ for the respective computable e).*

We prove the lemma in the rest of this section. We assume a given grammar $\mathcal{G} = (\mathcal{N}, \Sigma, \mathcal{R})$ and a term E_0 that is bisim-infinite.

An infinite simple-stair sequence witnessing bisim-infiniteness. Let us fix an infinite path $E_0 \xrightarrow{r_1} E_1 \xrightarrow{r_2} E_2 \xrightarrow{r_3} \cdots$ in $\mathcal{L}^{\text{R}}_{\mathcal{G}}$ such that $E_i \not\sim E_j$ (in $\mathcal{L}^{\text{A}}_{\mathcal{G}}$) for all $i \neq j$ (recall Prop. 3); this entails that there is *no repeat*, i.e., we have $E_i \neq E_j$ for all $i \neq j$. Hence there must be the least $i_0 \in \mathbb{N}$ such that $r_{i_0+1}r_{i_0+2}\ldots r_{i_0+\ell}$ is a stair for each $\ell \in \mathbb{N}$. (This obviously holds even if E_0 is an infinite term, since it has only finitely many subterms due to its regularity.) Moreover, given i_j, there must be the least i_{j+1} such that $i_j < i_{j+1}$ and $r_{i_{j+1}+1}r_{i_{j+1}+2}\ldots r_{i_{j+1}+\ell}$ is a stair for each $\ell \in \mathbb{N}$.

For each $j \in \mathbb{N}$ we put $H_j = E_{i_j}$, and we present the suffix of the above path starting with E_{i_0} as

$$H_0 \xrightarrow{w_1} H_1 \xrightarrow{w_2} H_2 \xrightarrow{w_3} \cdots, \text{ denoting } H_j = A_j(x_1,\ldots,x_m)\sigma_j\sigma_{j-1}\cdots\sigma_0 \tag{2}$$

where $A_j(x_1,\ldots,x_m) \xrightarrow{w_{j+1}} A_{j+1}(x_1,\ldots,x_m)\sigma_{j+1}$. We also write $H_j = A_j(x_1,\ldots,x_m)\sigma'_j$.

The words w_i are stairs; we can even assume that w_i are direct stairs (i.e., we replace them with the respective direct stairs if they are not direct stairs) while keeping the property that $H_i \not\sim H_j$ for all $i \neq j$. We thus assume that w_i are *direct stairs*, which in our case obviously implies that they are *simple stairs*.

A specific "keep-and-drown task" extracted from the bisim-infinite stair path. Prop. 10 captures a first step in demonstrating the existence of a simple witness induced by the path $H_0 \xrightarrow{w_1} H_1 \xrightarrow{w_2} H_2 \xrightarrow{w_3} \cdots$ (2), related to a fixed grammar $\mathcal{G} = (\mathcal{N}, \Sigma, \mathcal{R})$.

A *task* T is a tuple $(A, \textsc{Drown}, \textsc{keep})$ where $A \in \mathcal{N}$, $\textsc{Drown} \subseteq \{x_1,\ldots,x_m\}$, and $\textsc{keep} \in [1,m]$, $x_{\textsc{keep}} \in \textsc{Drown}$. We put $\textsc{NotCare} = \{x_1,\ldots,x_m\} \smallsetminus \textsc{Drown}$.

For a term G, by $G \models (\textsc{Drown}, z)$, for $z \in \mathbb{N} \cup \{\omega\}$, we denote that the depth of each occurrence of $x_i \in \textsc{Drown}$ in G (if there is any) is at least z, which means that x_i does not occur in G when $z = \omega$. By $G \models \textsc{keep}$ we denote that $x_{\textsc{keep}}$ occurs in G, and $G \models (\textsc{Drown}, z) \wedge \textsc{keep}$ denotes that we have $G \models (\textsc{Drown}, z)$ and $G \models \textsc{keep}$.

A *task* $\mathrm{T} = (A, \textsc{Drown}, \textsc{keep})$ is *satisfied for* $k \in \mathbb{N}$, which is denoted by $\models (\mathrm{T}, k)$, if there is $w \in \mathcal{R}^*$ and G such that $A(x_1, \dots, x_m) \xrightarrow{w} G$ and $G \models (\textsc{Drown}, k) \wedge \textsc{keep}$. By $\models (\mathrm{T}, \omega)$ we denote that we have $\models (\mathrm{T}, k)$ for all $k \in \mathbb{N}$.

A technical proof of the next proposition is given in the full arxiv-version.

▶ **Proposition 10.** *There is a task* $\mathrm{T} = (A, \textsc{Drown}, \textsc{keep})$ *and an infinite subsequence* $\textsc{Seq}$ *of the sequence* $0, 1, 2, \dots$ *such that* $\models (\mathrm{T}, \omega)$ *and the following conditions hold in the path (2), where we use the notation* $H_j = A_j(x_1, \dots, x_m)\sigma'_j$*:*
1. $A_j = A$ *for all* $j \in \textsc{Seq}$*;*
2. *for each* $x_i \in \textsc{NotCare}$ *we have* $x_i\sigma'_{j_1} \sim x_i\sigma'_{j_2}$ *for all* $j_1, j_2 \in \textsc{Seq}$*;*
3. *for each* $x_i \in \textsc{Drown}$ *we have* $x_i\sigma'_{j_1} \not\sim x_i\sigma'_{j_2}$ *for any* $j_1 \neq j_2$ *where* $j_1, j_2 \in \textsc{Seq}$*.*

Witness schemes. For our fixed path $H_0 \xrightarrow{w_1} H_1 \xrightarrow{w_2} H_2 \xrightarrow{w_3} \cdots$ (2) we aim to show that there are a stair u and an eligible stair w such that $H_0 \xrightarrow{u}\xrightarrow{w}$ is a simple witness (of bisim-infiniteness of H_0 and thus also of E_0). We will also have that both u and w are sequences of simple stairs, hence $uw = w_1 w_2 \cdots w_\ell$ where $A_0(x_1, \dots, x_m) \xrightarrow{w_1}\xrightarrow{w_2} \cdots \xrightarrow{w_\ell}$, each w_i is a simple stair, and the sequence $w = w_j w_{j+1} \dots w_\ell$ is *marked as a pumping stair*. It is useful to make the following generalization (of simple witnesses).

A *stair-scheme*, or just a *scheme* for short, is a sequence $W = w_1, w_2, \dots, w_\ell$ of (compatible) simple stairs, where $A(x_1, \dots, x_m) \xrightarrow{w_1}\xrightarrow{w_2} \cdots \xrightarrow{w_\ell}$ for A determined by the first grammar-rule r_1 in w_1, and where any segment $w_i w_{i+1} \dots w_j$ that is an eligible stair might be marked as a *pumping stair*; the pumping stairs can be "nested", one can be contained in another, but no pumping stair can start or end inside another pumping stair.

We use the notation of regular expressions with concatenation and iteration (star) to denote such schemes; an example is $u_1((v_1)^* u_2 (v_2)^*)^* u_3 (v_3)^* u_4 u_5 ((v_5)^* u_6)^*$ (where we have six pumping stairs, namely v_1, v_2, $v_1 u_2 v_2$, v_3, v_5, and $v_5 u_6$).

For a scheme W (like above), by $\textsc{Pump}(W, z)$, where $z \in \mathbb{N} \cup \{\omega\}$, we denote the sequence arising from $w_1 w_2 \dots w_\ell$ by repeating each pumping stair z times. (In our example, $\textsc{Pump}(W, z)$ is $u_1((v_1)^z u_2 (v_2)^z)^z u_3 (v_3)^z u_4 u_5 ((v_5)^z u_6)^z$.) In the case $z = \omega$ we get infinite "words" whose ordinal lengths can be bigger than ω, but since all pumping stairs are eligible, we can soundly define the terms $G_{(W,z)}$ by

$$A(x_1, \dots, x_m) \xrightarrow{\textsc{Pump}(W,z)} G_{(W,z)}; \text{ we also put } \textsc{Lim}_W = G_{(W,\omega)}.$$

We say that a scheme W, where A is the left-hand side nonterminal of the first rule in W, is a ("non-simple") *witness* (of bisim-infiniteness) *for* $A(x_1, \dots, x_m)\sigma$ if $G_{(W,k)}\sigma \not\sim \textsc{Lim}_W\sigma$ for infinitely many $k \in \mathbb{N}$.

It is not difficult to generalize Lemma 8 for schemes (viewed as candidates for witnesses), and to derive the next proposition (as is shown in the full version).

▶ **Proposition 11.** *A term* E_0 *has a simple witness (of bisim-infiniteness) iff there is a term* $H = A(x_1, \dots, x_m)\sigma$ *reachable from* E_0 *for which there is (a scheme* W *that is) a witness.*

It suffices that $\models (\mathrm{T}, \omega)$ can be demonstrated by a scheme. We first show that Lemma 12 suffices for finishing the proof of Theorem 1, and then we sketch a proof idea for the lemma.

▶ **Lemma 12.** *For any task* $\mathrm{T} = (A, \textsc{Drown}, \textsc{keep})$ *where* $\models (\mathrm{T}, \omega)$ *there is a scheme* W*, starting from* $A(x_1, \dots, x_m)$*, such that* $G_{(W,k)} \models \textsc{keep}$ *for all* $k \in \mathbb{N}$ *and* $G_{(W,\omega)} \models (\textsc{Drown}, \omega)$*; for such* W *we have* $G_{(W,k)} \models (\textsc{Drown}, k) \wedge \textsc{keep}$ *for all* $k \in \mathbb{N}$*.*

We consider a task $\mathrm{T} = (A, \textsc{Drown}, \textsc{keep})$, where $\models (\mathrm{T}, \omega)$, that can be extracted from the path $H_0 \xrightarrow{w_1} H_1 \xrightarrow{w_2} H_2 \xrightarrow{w_3} \cdots$ (2) by Prop. 10; we also recall the respective sequence $\textsc{Seq}$ and the notation $H_j = A(x_1, \ldots, x_m)\sigma'_j$. Let W be a scheme guaranteed by Lemma 12 for T; we recall the notation $\textsc{Lim}_W = G_{(W,\omega)}$. For all $j \in \textsc{Seq}$ the terms $\textsc{Lim}_W\, \sigma'_j$ are from the same bisim-class (by 2 in Prop. 10); let $\textsc{Lim}$ be a term representing this class.

For the sake of contradiction we now suppose that W is not a witness for any $H_j = A(x_1, \ldots, x_m)\sigma'_j$. Then there is some $e \in \mathbb{N}$ (determined by $\textsc{Lim}$) such that $G_{(W,e)}\sigma'_j \sim \textsc{Lim}$ for all $j \in \textsc{Seq}$ (due to the mentioned generalization of Lemma 8). Since there is $d \in \mathbb{N}$ such that $x_{\textsc{keep}}\sigma'_j \in \textsc{Region}(G_{(W,e)}\sigma'_j, d)$ for all $j \in \textsc{Seq}$, all bisim-classes $[x_{\textsc{keep}}\sigma'_j]_\sim$ for $j \in \textsc{Seq}$ must be in $\textsc{Region}([\textsc{Lim}]_\sim, d)$ in the quotient-LTS $(\mathcal{L}^{\mathrm{A}}_{\mathcal{G}})_\sim$ (which follows from the fact $G_{(W,e)}\sigma'_j \sim \textsc{Lim}$). There are thus only finitely many bisim-classes $[x_{\textsc{keep}}\sigma'_j]_\sim$ where $j \in \textsc{Seq}$, which contradicts with the condition 3 of Prop. 10 that $x_{\textsc{keep}}\sigma'_{j_1} \not\sim x_{\textsc{keep}}\sigma'_{j_2}$ for any $j_1 \neq j_2$ in $\textsc{Seq}$ (recall that $x_{\textsc{keep}} \in \textsc{Drown}$).

Hence W is a witness for some $H_j = A(x_1, \ldots, x_m)\sigma'_j$; by Prop. 11 this proves Lemma 9 (and thus Theorem 1).

The fact $\models (\mathrm{T}, \omega)$ can be demonstrated by a scheme. We now sketch a proof idea for Lemma 12. If $\models (\mathrm{T}, \omega)$, where $\mathrm{T} = (A, \textsc{Drown}, \textsc{keep})$, then there is a collection of paths $A(x_1, \ldots, x_m) \xrightarrow{w_k} G_k$ where $G_k \models (\textsc{Drown}, k) \wedge \textsc{keep}$, for all $k \in \mathbb{N}$. We can choose shortest possible words w_k; in fact, they are sequences of simple stairs.

Each path $A(x_1, \ldots, x_m) \xrightarrow{w_k} G_k$ must be progressing to its goal, stepwise "drowning" the (occurrences of) variables $x_i \in \textsc{Drown}$, while keeping at least one occurrence of $x_{\textsc{keep}}$. For a term F we can define its *drown-quality* as the function $\mathrm{DQ}(F) : \textsc{Drown} \to \mathbb{N} \cup \{\omega\}$ where $\mathrm{DQ}(F)(x_i)$ is the smallest (shallowest) depth of an occurrence of x_i in F (where $\mathrm{DQ}(F)(x_i) = \omega$ means that x_i does not occur in F). The *keep-quality* $\mathrm{KQ}(F)$ is one bit (1 ir 0) that captures the fact if $x_{\textsc{keep}}$ occurs in F. For each term $H = B(F_1, \ldots, F_m)$ on a path $A(x_1, \ldots, x_m) \xrightarrow{w_k} G_k$ we define its *level-quality* as $\mathrm{LQ}(H) = (B, \mathrm{DQ}(F_1), \ldots, \mathrm{DQ}(F_m), \mathrm{KQ}(F_1), \ldots, \mathrm{KQ}(F_m))$.

By standard facts, in particular Dickson's Lemma and König's Lemma, in any sufficiently long w_k there is an eligible stair that keeps or increases the level-quality in each component (where we put $B \leq B'$ if $B = B$). This does not solve the problem completely, due to the possible long segments with root-sticking depth-1 subterms. This subtle point is handled in the full arxiv-version.

4 Additional Remarks

The mentioned deterministic case studied by Valiant [13] could be roughly explained as follows: for a deterministic grammar, if an eligible stair is reachable from E_0 where the start and the end of the stair are non-equivalent, then E_0 is bisim-infinite. Hence by compositionality a bound on the size of the potential equivalent finite system can be derived, and thus decidability of the full equivalence is not needed here.

In the case equivalent to *normed* pushdown processes, the regularity problem essentially coincides with the boundedness problem, and is thus much simpler. (See, e.g., [11] for a further discussion.)

Acknowledgements. This work has been supported by the Grant Agency of the Czech Rep., project GAČR:15-13784S. I also thank Stefan Göller for drawing my attention to the decidability question for regularity of pushdown processes, and for discussions about some related works (like [13]).

References

1. Michael Benedikt, Stefan Göller, Stefan Kiefer, and Andrzej S. Murawski. Bisimilarity of pushdown automata is nonelementary. In *Proc. LICS 2013*, pages 488–498. IEEE Computer Society, 2013.
2. Christopher H. Broadbent and Stefan 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 für Informatik, 2012.
3. Bruno Courcelle. Recursive applicative program schemes. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science, vol. B*, pages 459–492. Elsevier, MIT Press, 1990.
4. Petr Jančar. Bisimulation equivalence of first-order grammars. In *Proc. ICALP'14 (II)*, volume 8573 of *LNCS*, pages 232–243. Springer, 2014.
5. Petr Jančar. Equivalences of pushdown systems are hard. In *Proc. FOSSACS 2014*, volume 8412 of *LNCS*, pages 1–28. Springer, 2014.
6. Petr Jančar and Jiri Srba. Undecidability of bisimilarity by defender's forcing. *J. ACM*, 55(1), 2008. `doi:10.1145/1326554.1326559`.
7. Antonín Kučera and Richard Mayr. On the complexity of checking semantic equivalences between pushdown processes and finite-state processes. *Inf. Comput.*, 208(7):772–796, 2010.
8. Sylvain Schmitz. Complexity hierarchies beyond elementary. *TOCT*, 8(1):3, 2016.
9. Géraud Sénizergues. L(A)=L(B)? Decidability results from complete formal systems. *Theoretical Computer Science*, 251(1–2):1–166, 2001.
10. Géraud Sénizergues. The bisimulation problem for equational graphs of finite out-degree. *SIAM J.Comput.*, 34(5):1025–1106, 2005.
11. Jiri 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. Updated version at http://users-cs.au.dk/srba/roadmap/.
12. Colin Stirling. Deciding DPDA equivalence is primitive recursive. In *Proc. ICALP'02*, volume 2380 of *LNCS*, pages 821–832. Springer, 2002.
13. Leslie G. Valiant. Regularity and related problems for deterministic pushdown automata. *J. ACM*, 22(1):1–10, 1975. `doi:10.1145/321864.321865`.