Reachability analysis of first-order definable pushdown systems

Lorenzo Clemente[1] **and Sławomir Lasota**[1]

1 **University of Warsaw**

Abstract

We study pushdown systems where control states, stack alphabet, and transition relation, instead of being finite, are first-order definable in a fixed countably-infinite structure. We show that the reachability analysis can be addressed with the well-known saturation technique for the wide class of *oligomorphic structures*. Moreover, for the more restrictive *homogeneous structures*, we are able to give concrete complexity upper bounds. We show ample applicability of our technique by presenting several concrete examples of homogeneous structures, subsuming, with optimal complexity, known results from the literature. We show that infinitely many such examples of homogeneous structures can be obtained with the classical *wreath product* construction.

1998 ACM Subject Classification F.1.1 [*Computation by Abstract Devices*]: Models of Computation; F.2.2 [*Nonnumerical Algorithms and Problems*]: Computations on discrete structures; F.3.1 [*Specifying and Verifying and Reasoning about Programs*]: Mechanical verification; F.4.1 [*Mathematical Logic*]: Logic and constraint programming.

Keywords and phrases automata theory, pushdown systems, sets with atoms, saturation technique.

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

Context. Pushdown automata (PDS) are a well-known model of recursive programs, with applications in areas as diverse as language processing, data-flow analysis, security, computational biology, and program verification. Many interesting analyses reduce to checking reachability in the infinite configuration graph generated by a PDS, which can be done in PTIME with the popular *saturation algorithm* [6, 17] (cf. also the recent survey [10]). Saturation shows a slightly more general property of PDS graphs, which is sometimes called *preservation of regularity*: Given a PDS and a regular set of target configurations thereof, compute the set of all configurations which can reach the target in a finite number of steps. The procedure is *effective*, in the sense that given an NFA recognizing the target set, an NFA recognizing the predecessors can be computed in PTIME. This is a central theoretical result in the analysis of PDS, with immediate practical applications as demonstrated by the prominent tool MOPED [16]. Therefore, it is of interest to extend this conceptually simple and yet powerful method to more general settings.

Several generalizations of the pushdown structure yielding PDS-like models admitting effective preservation of regularity are known, e.g., tree-pushdown systems [19], ordered multi-pushdown systems [8, 4], annotated higher-order pushdown systems [24, 9], and strongly normed multi-pushdown systems [13]. In this paper, instead of generalizing the pushdown structure itself, we generalize the *contents* of the pushdown, by allowing the pushdown symbols to be drawn from an infinite set. Our model is parametric in the choice of a countably-infinite logical structure $\mathbb{A}$, called *atoms*. We introduce and study *first-order*

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

definable pushdown systems (FO-definable PDS) over $\mathbb{A}$, which are like usual PDS, except that control locations, stack alphabet, and transition relation are FO-definable sets over $\mathbb{A}$, instead of ordinary finite sets. Thus, we do not invent a new model, but we reinterpret the classical model in a new setting. This covers ordinary PDS as a special case, and allows the study of non-trivial yet decidable classes of PDS over infinite alphabets. For instance, by taking $\mathbb{A}$ to be *equality atoms* $(\mathbb{D}, =)$, i.e., a countably-infinite set $\mathbb{D}$ where only equality testing is allowed, we obtain (and slightly generalize) pushdown register automata [11, 25].

Contributions and organization. The technical results of this paper and its structure are as follows. In Sec. 2, we introduce the setting of FO-definable sets, FO-definable relations, and FO-definable NFA. In Sec. 3, we introduce FO-definable PDS. This is done by reinterpreting the classical model in the FO-definable framework. Our approach has the advantage that we do not need to define a new model. Instead, we *reinterpret* the classical model in a generic logical framework. In Sec. 4, we consider *oligomorphic atoms*[1] with a decidable first-order theory, and we show effective preservation of regularity for the backward reachability relation of configuration graphs of FO-definable PDS. This is obtained via a symbolic implementation of the classical saturation method, which comes along with a simple proof of correctness. In Sec. 5, we provide an upper bound in the special case of *homogeneous* atoms, and in particular an `ExpTime` upper complexity bound in the case of *tractable* homogenous atoms, matching the known `ExpTime`-hardness for equality atoms from [25]. In Sec. 6, we provide many interesting examples of tractable homogeneous atoms for which we can apply our results, including equality atoms [25] (as remarked above), but also: *total order atoms* $(\mathbb{Q}, \leq)$, which can be used for modeling densely-ordered data values; *equivalence atoms* $(\mathbb{D}, R)$, where R is an equivalence relation of infinite index s.t. each equivalence class is infinite, which can be used to model nested data values; *universal tree atoms*, which can be used to model dynamic topologies of concurrent programs with process creation and termination; as well as other structures, such as *universal partial order atoms*, *universal tournament atoms*, and *universal graph atoms* [23]. In the same section, we also show that the classic *wreath product* construction can be used to generate infinitely many such tractable examples. Our logical approach has the advantage to highlight the general principle behind decidability, and we can thus prove correctness once and for all for *all* structures satisfying the mild assumptions above. As a byproduct, we also obtain tight complexity results for PDS over natural classes of infinite alphabets. Infinitely many such natural structures can be found by using the wreath product construction. In Sec. 7 we conclude with some directions for future work.

2 Preliminaries

Sets with atoms. Let $\mathbb{A}$ be a countably-infinite logical structure with finite vocabulary. An element of the structure we call *atom*, and the whole structure we call *atoms*. Examples of atoms are equality atoms $(\mathbb{D}, =)$, i.e., an arbitrary countable infinite set $\mathbb{D}$ with equality, and total order atoms $(\mathbb{Q}, \leq)$, i.e., the rationals with the dense order. More examples of atoms will be discussed in Sec. 6. In the study of atoms, it plays a central role the group $\mathrm{Aut}(\mathbb{A})$ of automorphisms of $\mathbb{A}$. For instance, automorphisms of equality atoms are all permutations of $\mathbb{D}$, and automorphisms of total order atoms are monotonic permutations of $\mathbb{Q}$. By using atoms, we can build sets containing either previously built sets, or atoms themselves. For example, we build tuples $\mathbb{A}^n$ of fixed length, or disjoint unions thereof. On such sets, we will consider

[1] A structure $\mathbb{A}$ is *oligomorphic* if for every n, the product $\mathbb{A}^n$ is orbit-finite.

the natural action of $\mathrm{Aut}(\mathbb{A})$, which renames atoms while keeping intact the remaining structure. For instance, on tuples of atoms the natural action is the point-wise renaming: for $\pi \in \mathrm{Aut}(\mathbb{A})$ and $a_1, \ldots, a_n \in \mathbb{A}$, $\pi(a_1, \ldots, a_n) = (\pi(a_1), \ldots, \pi(a_n))$. Similarly, on disjoint unions the action is component-wise. The action induces the notion of *orbit*, which is the set of elements that can be reached via renaming, i.e., $\mathrm{orbit}(e) = \{\pi(e) \mid \pi \in \mathrm{Aut}(\mathbb{A})\}$. The sets in the sequel will always be *equivariant*, i.e., invariant under action of automorphisms[2]. Every orbit is equivariant by definition, and every equivariant set is a disjoint union of orbits. For instance, in total order atoms $(\mathbb{Q}, \leq)$, the set $\mathbb{Q}^2$ is the disjoint union of 3 orbits, $\{(q, q') \mid q < q'\}$, $\{(q, q') \mid q = q'\}$, and $\{(q, q') \mid q > q'\}$; and $\mathbb{Q}^2 \uplus \mathbb{Q}^3$ is the disjoint union of 16 orbits. A central notion is that of *orbit-finite* sets, which are *finite* unions of orbits (as opposed to arbitrary unions). Intuitively, an orbit-finite set has only finitely many elements up to renaming by atom automorphisms. Orbit-finiteness generalizes finiteness, and a substantial portion of results from automata theory carry over to the more general orbit-finite setting [5]. This paper can be seen as such a case study for the specific case of pushdown automata. For the sake of concreteness, we restrict in the rest of the paper to *FO-definable sets*, to be defined now; we only note that the results of this paper can be straightforwardly generalized to all orbit-finite sets with atoms.

FO-definable sets. Fix a structure $\mathbb{A}$ over a finite vocabulary. We describe infinite sets symbolically using first-order logic over the vocabulary of $\mathbb{A}$, which we assume to always include the equality relation $=$. A first-order formula $\varphi(\vec{x})$ (where we explicit list all free variables according to an implicit order) *defines* the subset $[\varphi] \subseteq \mathbb{A}^n$ of tuples that satisfy φ, i.e., $[\varphi] = \{\vec{a} \in \mathbb{A}^n \mid (\vec{x} \mapsto \vec{a}) \vDash \varphi\}$. This set is always equivariant. The *dimension* of $[\varphi]$ is the number n of free variables of φ, denoted by $\dim \varphi$; as a limit case, when $\dim \varphi = 0$, $[\varphi]$ is a singleton. A *FO-definable set X over* $\mathbb{A}$ is a finite indexed union of such sets, i.e.,

$$X = \bigcup_{l \in L} \{l\} \times [\varphi_l], \qquad \text{where } L \text{ is a finite index set.}$$

When we want to omit the formal indexing, we just write X as the finite disjoint union $\biguplus_{l \in L} [\varphi_l]$. Since FO-definable sets are unions of equivariant sets, they are equivariant too. When $\dim \varphi_l = 0$ for every $l \in L$, then X is finite and has the same number of elements as L. Thus, FO-definable sets generalize finite sets.

We use FO-definable sets for control locations and alphabets of automata. In the former case, an index $l \in L$ may be understood as a control location, and a tuple $\vec{a} \in \mathbb{A}^n$ as a valuation of n registers. Under this intuition, φ_l is an invariant that constraints register valuations in a control location l. We do not assume that all component sets $[\varphi_l]$ have the same dimension, i.e., the number of registers may vary from one control location to another.

FO-definable relations. Along the same lines, we define FO-definable binary relations. Consider two FO-definable sets $X = \biguplus_{l \in L} [\varphi_l]$ and $Y = \biguplus_{k \in K} [\psi_k]$. An FO-definable relation $R \subseteq X \times Y$ is an FO-definable set $R = \biguplus_{l \in L, k \in K} [\xi_{lk}]$ where the indexing set is the Cartesian product $L \times K$, and every component set $[\xi_{lk}]$ satisfies $[\xi_{lk}] \subseteq [\varphi_l] \times [\psi_k]$. In particular, $\dim \xi_{lk} = \dim \varphi_l + \dim \psi_k$. Relations of greater arities can be obtained by iterating the construction above. We use FO-definable relations to define transition relations of automata.

[2] More generally, one can consider *finitely supported* sets. A set is supported by $S \subseteq_{\mathrm{fin}} \mathbb{A}$ if it is invariant under automorphisms that preserve elements of S. The results of this paper can be straightforwardly generalized to finitely supported sets.

The formula ξ_{lk} may be understood as a constraint on a transition from control location l to control location k, prescribing how a valuation of registers in l before the transition relates to a valuation of registers in k after the transition.

FO-definable NFA. As an example application of FO-definable sets and relations, we define FO-definable NFA. This model will be used later to recognize regular set of configurations of FO-definable PDS, also defined later. A classical NFA is a tuple $\mathcal{A} = (\Gamma, Q, F, \delta)$, where Γ is a finite input alphabet, Q is a finite set of states, of which those in $F \subseteq Q$ are the final ones, and $\delta \subseteq Q \times \Gamma \times Q$ is the transition relation. Once an initial state is chosen, the definitions of run, accepting run, and language $\mathcal{L}(\mathcal{A})$ recognized by $\mathcal{A}$ are standard. By simply replacing "finite" with "FO-definable" in the definition above, we obtain FO-definable NFA. To fix notation, a FO-definable NFA will be written as a tuple $\mathcal{A} = (\Gamma = \biguplus_{k\in K}[\varphi_k],\ Q = \biguplus_{l\in L}[\psi_l],\ F = \biguplus_{l\in L}[\psi_l^F],\ \delta = \biguplus_{l,l'\in L,k\in L}[\delta_{lkl'}])$, where w.l.o.g. we assume that Q and F have the same index set L. Notice that δ is a FO-definable set, while $\delta_{lkl'}$ is a first-order formula.

▶ **Example 1.** Let $\mathbb{A}$ be the total order atoms $(\mathbb{Q}, \leq)$, and let the alphabet be $\Gamma = \{k\} \times \mathbb{Q}$. Consider the language $M = \{(k, a_1) \cdots (k, a_n) \in \Gamma^* \mid a_1 \geq a_2 \leq a_3 \geq \cdots \leq a_{2n+1}\}$ of non-empty finite words of odd length of alternating growth. This language can be recognized from state ℓ_I by the NFA

$$\mathcal{A} = (\Gamma,\ Q = \{\ell_I\} \cup \{\ell_0\} \times \mathbb{Q} \cup \{\ell_1\} \times \mathbb{Q}, F = \{\ell_0\} \times \mathbb{Q}, \delta = \biguplus_{l,l'\in\{\ell_I,\ell_0,\ell_1\}} [\delta_{lkl'}]).$$

The initial location ℓ_I does not contain any register, while control locations ℓ_0, ℓ_1 both contain one register, which is used to guess the next input symbol and to ensure the right ordering. Formally, $\delta_{\ell_I k \ell_0}(, y, x') \equiv x' \leq y$ (we use the notation $\delta_{\ell_I k \ell_0}(, y, x')$ to emphasize that ℓ_I does not have any register), $\delta_{\ell_0 k \ell_1}(x, y, x') \equiv (x = y \wedge x' \geq y)$, $\delta_{\ell_1 k \ell_0}(x, y, x') \equiv (x = y \wedge x' \leq y)$, and $[\delta_{lkl'}] = \emptyset$ for the other cases.

3 First-order definable pushdown systems

In this section we define FO-definable PDS and their reachability problem. According to the classical definition, a pushdown system (PDS) $\mathcal{P} = \langle \Gamma, P, \rho \rangle$ consists of a finite stack alphabet Γ, a finite set of control states P, and a finite set of transition rules $\rho = \rho^{\mathsf{push}} \cup \rho^{\mathsf{pop}}$, which is partitioned into push rules $\rho^{\mathsf{push}} \subseteq P \times \Gamma \times P \times \Gamma \times \Gamma$ and pop rules $\rho^{\mathsf{pop}} \subseteq P \times \Gamma \times P$. In this paper, we reinterpret this definition in the setting of FO-definable sets, which yields a more general model. For an atom structure $\mathbb{A}$, *FO-definable PDS over* $\mathbb{A}$ are obtained by replacing "finite set" with "FO-definable set" in the classical definition. To fix notation, a FO-definable PDS is a tuple

$$\mathcal{P} = \langle \Gamma = \biguplus_{k\in K} [\varphi_k],\ P = \biguplus_{\ell\in L} [\xi_\ell],\ \rho = \rho^{\mathsf{push}} \cup \rho^{\mathsf{pop}} \rangle,$$

where $\rho^{\mathsf{push}} = \biguplus_{\ell,\ell'\in L,k,k',k''\in K} [\rho^{\mathsf{push}}_{\ell k \ell' k' k''}]$ and $\rho^{\mathsf{pop}} = \biguplus_{\ell,\ell'\in L,k\in K} [\rho^{\mathsf{pop}}_{\ell k \ell'}]$. As in the classical case, a FO-definable PDS induces an infinite transition system $\langle \mathcal{C}, \longrightarrow \rangle$, where the set of configurations is $\mathcal{C} = P \times \Gamma^*$, and there is a transition $c \longrightarrow c'$ between two configurations $c = (q, aw)$ and $c' = (q', w')$ if, and only if, either there exists a push rule $(q, a, q', b, c) \in \rho^{\mathsf{push}}$ s.t. $w' = bcw$, or there exists a pop rule $(q, a, q') \in \rho^{\mathsf{pop}}$ s.t. $w = w'$. Let $\longrightarrow^*$ be the reflexive

and transitive closure of $\longrightarrow$. For a set C of configurations, the *backward reachability set of* C, denoted $\mathrm{Reach}_{\mathcal{P}}^{-1}(C)$, is the set of configurations that can reach some configuration in C:

$$\mathrm{Reach}_{\mathcal{P}}^{-1}(C) = \{c \in \mathcal{C} \mid c \longrightarrow^* c' \text{ for some } c' \in C\} \ .$$

▶ **Example 2.** We define a FO-definable PDS $\mathcal{P}$ over total order atoms $(\mathbb{Q}, \leq)$ which constructs strictly monotonic stacks, the maximal element being on the top of the stack. Let $\mathcal{P} = \langle \Gamma = \{k\} \times \mathbb{Q}, P = \{\ell_I\}, \rho = \rho^{\mathsf{push}} \rangle$, where $\rho^{\mathsf{push}}_{\ell_I k \ell_I k k}(, y, , y', y'') \equiv (y < y' \wedge y'' = y)$.

This paper concentrates on the reachability analysis for FO-definable PDS. Given a FO-definable PDS $\mathcal{P} = \langle \Gamma, P, \rho \rangle$, two control locations $p, q \in P$, and a stack symbol $\bot \in \Gamma$, the *reachability problem* asks whether $(p, \bot) \in \mathrm{Reach}_{\mathcal{P}}^{-1}(\{q\} \times \Gamma^*)$. We start with stack $\bot$ and we ignore the stack at the end of the computation. More general analyses can be considered by imposing regular constraints on the initial and final stack contents. These easily reduce to reachability of a regular set of configurations, which is the problem considered in the next section.

4 Preservation of regularity I: Oligomorphic atoms

We solve the reachability problem as a corollary of a general effective preservation of regularity result for the backward reachability relation of FO-definable PDS. To this end, we use FO-definable NFA to describe regular sets of configurations. In the following, fix a FO-definable PDS $\mathcal{P} = \langle \Gamma, P, \rho \rangle$, and a FO-definable NFA $\mathcal{A} = \langle \Gamma, Q, F, \delta \rangle$ s.t. $P \subseteq Q$. The NFA $\mathcal{A}$ recognizes the following language $L_{\mathcal{P}}(\mathcal{A})$ of configurations of $\mathcal{P}$,

$$\mathcal{L}_{\mathcal{P}}(\mathcal{A}) = \{(p, w) \in P \times \Gamma^* \mid \mathcal{A} \text{ accepts } w \text{ from state } p\} .$$

Such sets of configurations of $\mathcal{P}$ we call *regular*. We assume w.l.o.g. that states of $\mathcal{A}$ that belong to P do not have incoming transitions, i.e. $\delta \subseteq Q \times \Gamma \times (Q \setminus P)$.

▶ **Example 3.** Recall the FO-definable PDS $\mathcal{P}$ from Example 2 building strictly monotonic stacks (maximal element on top). Let N be the following set of configurations

$$N = \{(\ell_I, (k, a_1) \cdots (k, a_{2n+1})) \in P \times \Gamma^* \mid a_1 \geq a_2 \leq a_3 \geq \cdots \leq a_{2n+1}\} .$$

This set is regular, and it is recognized by the NFA $\mathcal{A}$ from Example 1, i.e., $\mathcal{L}_{\mathcal{P}}(\mathcal{A}) = N$. The backward reachability set is

$$\mathrm{Reach}_{\mathcal{P}}^{-1}(N) = N \cup \{(\ell_I, (k, a_2) \cdots (k, a_{2n+1})) \in P \times \Gamma^* \mid a_2 \leq a_3 \geq \cdots \leq a_{2n+1}\} .$$

We will see below how to compute a FO-definable NFA recognizing $\mathrm{Reach}_{\mathcal{P}}^{-1}(N)$.

We solve the reachability problem for PDS over *oligomorphic* atoms.[3]. Oligomorphicity is an important notion in model theory [23]. Formally, a structure is oligomorphic if, and only if, for every $n \in \mathbb{N}$, the set $\mathbb{A}^n$ is orbit-finite. Not all structures are oligomorphic, as shown in the following example.

[3] One could also consider PDS defined by general prefix rewriting, i.e., with transitions in $\rho \subseteq P \times \Gamma^* \times P \times \Gamma^*$. For oligomorphic atoms, our simplified push/pop model can simulate prefix rewriting while preserving reachability properties (but not configuration graph isomorphism, or even bisimilarity), like in the classical case.

(0) $\delta' := \delta \cup \rho^{\text{pop}}$

(1) `repeat`

(2) $\quad \delta' := \delta' \cup \text{forced}(\delta')$

(3) `until` $\text{forced}(\delta') \subseteq \delta'$

Figure 1 Abstract saturation algorithm.

▶ Remark (Timed atoms). Timed atoms $(\mathbb{Q}, \leq, +1)$ is a well-known example of non-oligomorphic structure. They extend total order atoms $(\mathbb{Q}, \leq)$ with the successor relation $(+1) \subseteq \mathbb{Q} \times \mathbb{Q}$. Automorphisms of timed atoms are monotone bijections π of $\mathbb{Q}$ that preserve unit intervals, i.e., $\pi(x+1) = \pi(x) + 1$. To see why timed atoms are non oligomorphic, it suffices to see that already $\mathbb{Q}^2$ has infinitely-many orbits. Indeed, for each $z \in \mathbb{Z}$, Q^2 has a disjoint orbit $\{(x, y) \in Q^2 \mid x - y = k\}$. (Since automorphisms preserve unit intervals, they preserve all integer distances.) Working in non-oligomorphic structures like timed atoms requires the use of specialized techniques, and the generic algorithm presented in this section does not terminate. We have thoroughly studied the reachability problem for FO-definable pushdown systems and automata over timed atoms in [12].

Since oligomorphic atoms are very general, we can merely state decidability of the reachability problem, without any complexity bounds. The only additional assumption that we require is decidability of the *first-order satisfiability problem* in the structure $\mathbb{A}$, which asks, given a first-order formula $\varphi(x_1, \ldots, x_n)$, whether some valuation $\eta : \{x_1, \ldots, x_n\} \to \mathbb{A}$ of its free variables satisfies φ.

▶ **Theorem 4.** *Let $\mathbb{A}$ be an oligomorphic structure with a decidable first-order satisfiability problem. For FO-definable PDS $\mathcal{P}$ over $\mathbb{A}$ and a FO-definable NFA $\mathcal{A}$ over $\mathbb{A}$ recognizing a regular set of configurations $L_{\mathcal{P}}(\mathcal{A})$, one can effectively construct a FO-definable NFA $\mathcal{B}$ over $\mathbb{A}$ recognizing $L_{\mathcal{P}}(\mathcal{B}) = Reach_{\mathcal{P}}^{-1}(L_{\mathcal{P}}(\mathcal{A}))$.*

We prove Theorem 4 by using the classical *saturation technique* [6, 17]. We first describe a simple abstract algorithm manipulating infinite sets of transitions, and then we show how this can be implemented symbolically at the level of formulas. As in the classical case, the FO-definable NFA $\mathcal{B}$ which is computed by the algorithm is of the form $\langle \Gamma, Q, F, \delta' \rangle$ with $\delta \subseteq \delta'$, i.e., it is obtained by adding certain transitions to $\mathcal{A}$. For any relation $\alpha \subseteq Q \times \Gamma \times Q$, let $\text{forced}(\alpha) \subseteq Q \times \Gamma \times Q$ be the following set of triples:

$$\text{forced}(\alpha) = \left\{(q, a, q') \mid \exists (q, a, q'', b, c) \in \rho^{\mathsf{push}}, \exists (q'', b, q''') \in \alpha, \exists (q''', c, q') \in \alpha\right\}.$$

The abstract saturation algorithm is shown in Fig. 1. The algorithm is partially correct for every structure $\mathbb{A}$ (even though it might not terminate). This follows directly from the observation that the saturated NFA $\mathcal{B}$ has a transition $(q, a, q') \in \delta'$ between states $q, q' \in P$ of $\mathcal{P}$ if, and only if, $\mathcal{P}$ admits a run $(q, a) \longrightarrow^* (q', \varepsilon)$ (we use here the assumption that no transition of $\mathcal{A}$ ends in a state $q \in P$ of $\mathcal{P}$). However, on arbitrary structures saturation does not terminate, either because the inclusion checking on line (3) is not decidable, or because it never actually holds. The first issue is addressed by the requirement that $\mathbb{A}$ has a decidable first-order satisfiability problem, and the second one by the fact that $\mathbb{A}$ is an oligomorphic structure.

We implement the abstract algorithm from Fig. 1 symbolically, by manipulating formulas instead of actual transitions. We assume w.l.o.g. that the index set of P (the control locations

INPUT: a FO-definable PDS $\mathcal{P} = \langle \Gamma = \biguplus_k [\varphi_k], P = \biguplus_\ell [\xi_\ell], \rho^{\mathsf{push}} \cup \rho^{\mathsf{pop}} \rangle$, with

$$\rho^{\mathsf{push}} = \biguplus_{\ell k \ell' k' k''} [\rho^{\mathsf{push}}_{\ell k \ell' k' k''}], \rho^{\mathsf{pop}} = \biguplus_{\ell k \ell'} [\rho^{\mathsf{pop}}_{\ell k \ell'}], \text{ and a FO-definable NFA}$$

$$\mathcal{A} = \langle \Gamma, Q = \biguplus_\ell [\psi_\ell], \delta = \biguplus_{\ell k \ell'} [\delta_{\ell k \ell'}] \rangle, \text{ with } [\xi_\ell] \subseteq [\psi_\ell], \text{ for every } \ell \in L.$$

(0) for every ℓ, k, ℓ' : $\delta'_{\ell k \ell'}(\vec{x}, \vec{y}, \vec{x}') := \delta_{\ell k \ell'}(\vec{x}, \vec{y}, \vec{x}') \vee \rho^{\mathsf{pop}}_{\ell k \ell'}(\vec{x}, \vec{y}, \vec{x}')$

(1) `repeat`

(2) for every ℓ, k, ℓ' : $\delta'_{\ell k \ell'}(\vec{x}, \vec{y}, \vec{x}') := \delta'_{\ell k \ell'}(\vec{x}, \vec{y}, \vec{x}') \vee \mathrm{forced}(\delta')_{\ell k \ell'}(\vec{x}, \vec{y}, \vec{x}')$

(3) `until(` $\bigwedge_{\ell,k,\ell'} \forall \vec{x}, \vec{y}, \vec{x}' \cdot \mathrm{forced}(\delta')_{\ell k \ell'}(\vec{x}, \vec{y}, \vec{x}') \implies \delta'_{\ell k \ell'}(\vec{x}, \vec{y}, \vec{x}'))$

Figure 2 Concrete saturation algorithm; ℓ, ℓ' range over L, and k ranges over K.

of $\mathcal{P}$) is the same as the index set of Q (the states of $\mathcal{A}$). First, notice that the set $\mathrm{forced}(\alpha)$ is FO-definable whenever α is so, since it can be expressed as follows:

$$\mathrm{forced}(\alpha)_{\ell k \ell'}(\vec{x}, \vec{y}, \vec{x}') := \bigvee_{\ell'', \ell''' \in L, k', k'' \in K} \exists \vec{x}'', \vec{y}', \vec{y}'', \vec{x}''' \cdot \rho^{\mathsf{push}}_{\ell k \ell'' k' k''}(\vec{x}, \vec{y}, \vec{x}'', \vec{y}', \vec{y}'') \wedge \alpha_{\ell'' k' \ell'''}(\vec{x}'', \vec{y}', \vec{x}''') \wedge \alpha_{\ell''' k'' \ell'}(\vec{x}''', \vec{y}'', \vec{x}'),$$

where L is the index set of Q, and K is the index set of Γ. Steps (0) and (2) of the algorithm are implemented by disjunction, and the test (3) is computable whenever first order satisfiability is so. We obtain the concrete algorithm in Fig. 2. Termination is guaranteed since $\mathbb{A}$ is oligomorphic, which implies orbit-finiteness of $Q \times \Gamma \times Q$. Indeed, δ' is always a union of orbits at every stage (i.e., equivariant), and therefore at least one orbit is added to δ' at every iteration.

▶ **Example 5.** We apply the concrete saturation algorithm to the PDS $\mathcal{P}$ and NFA $\mathcal{A}$ from Example 3. Recall that $\mathcal{P} = \langle \Gamma = \{k\} \cup \mathbb{Q}, P = \{\ell_I\}, \rho^{\mathsf{push}} \rangle$, with $\rho^{\mathsf{push}}_{\ell_I k \ell_I k k}(, y, , y', y'') \equiv (y < y' \wedge y'' = y)$, and $\mathcal{A} = \langle \Gamma, Q = \{\ell_I\} \cup \{\ell_0, \ell_1\} \times \mathbb{Q}, F = \{\ell_0\} \times \mathbb{Q}, \delta \rangle$, with $\delta_{\ell_I k \ell_0}(, y, x') \equiv x' \leq y$, $\delta_{\ell_0 k \ell_1}(x, y, x') \equiv (x = y \wedge x' \geq y)$, $\delta_{\ell_1 k \ell_0}(x, y, x') \equiv (x = y \wedge x' \leq y)$ (omitting the trivial cases). For the first iteration, let $\delta^0 := \delta$. We compute $\mathrm{forced}(\delta^0)$, for which the only nontrivial case is $\mathrm{forced}(\delta^0)_{\ell_I k \ell_1}(, y, x') \equiv \exists y', y'', x''' \cdot \rho^{\mathsf{push}}_{\ell_I k \ell_I k k}(, y, , y', y'') \wedge \delta^0_{\ell_I k \ell_0}(, y', x''') \wedge \delta^0_{\ell_0 k \ell_1}(x''', y'', x')$, which equals

$$\exists y', y'', x''' \cdot (y < y' \wedge y'' = y) \wedge (x''' \leq y') \wedge (x''' = y'' \wedge x' \geq y'').$$

By removing quantifiers (thanks to the density of $\mathbb{Q}$), the former is equivalent to $x' \geq y$. Therefore, δ^1 extends δ^0 with the new transition $\delta^1_{\ell_I k \ell_1}(, y, x') \equiv (x' \geq y)$. Since δ^1 is not equivalent to δ^0, we go to the next iteration. We compute $\mathrm{forced}(\delta^1)$, for which the only new case is $\mathrm{forced}(\delta^1)_{\ell_I k \ell_0}(, y, x') \equiv \exists y', y'', x''' \cdot \rho^{\mathsf{push}}_{\ell_I k \ell_I k k}(, y, , y', y'') \wedge \delta^1_{\ell_I k \ell_1}(, y', x''') \wedge \delta^1_{\ell_1 k \ell_0}(x''', y'', x')$, which equals

$$\exists y', y'', x''' \cdot (y < y' \wedge y'' = y) \wedge (x''' \geq y') \wedge (x''' = y'' \wedge x' \leq y'').$$

The latter is equivalent to $\exists y' \cdot y < y' \wedge y \geq y' \wedge x' \leq y$, which is clearly unsatisfiable. Therefore δ^2 is equivalent to δ^1, and the algorithms stops. It is immediate to check that

$\mathcal{B} = \langle \Gamma, Q = \ell_I \cup \{\ell_0, \ell_1\} \times \mathbb{Q}, F = \{\ell_0\} \times \mathbb{Q}, \delta^1 \rangle$ recognizes precisely $\mathrm{Reach}_{\mathcal{P}}^{-1}(N)$, where $N = \mathcal{L}_{\mathcal{P}}(\mathcal{A})$.

5 Preservation of regularity II: Homogeneous atoms

Relational homogeneous structures are a well-behaved subclass of oligomorphic structures, for which we are able to give precise complexity upper bounds for our saturation construction. A relational structure $\mathbb{A}$ (i.e., with no function symbols in the vocabulary) is *homogeneous* if every isomorphism between two finite induced substructures of $\mathbb{A}$ extends to an automorphism of the whole $\mathbb{A}$. This immediately implies that $\mathbb{A}$ is oligomorphic.

▶ Proposition 1. Let $\mathbb{A}$ be a relational homogeneous structure. For $n \geq 1$, the number of orbits of $\mathbb{A}^n$ is bounded by $2^{\mathrm{poly}(n)}$.

Proof. A tuple of n elements $(a_1, \ldots, a_n) \in \mathbb{A}^n$ can be seen as an induced substructure of $\mathbb{A}$, where elements are additionally labelled with the positions $\{1 \ldots n\}$. Two such induced substructures $\bar{a}, \bar{b} \in \mathbb{A}^n$ are isomorphic exactly when the elements $\bar{a}$ and $\bar{b}$ satisfy the same relations in the vocabulary of $\mathbb{A}$. Therefore, there number of isomorphism classes is bounded by $2^{\mathrm{poly}(n)}$. Since $\mathbb{A}$ is homogeneous, every isomorphism between $\bar{a}$ and $\bar{b}$ extends to an automorphism of the whole $\mathbb{A}$, and thus $\bar{a}$ and $\bar{b}$ are in the same orbit. Consequently, the same bound applies to the number of orbits of $\mathbb{A}^n$. ◀

All structures listed in the introduction are homogeneous relational structures. However, not all oligomorphic relational structures are homogeneous, as the example below shows.

▶ **Example 6** (Bit vector atoms)**.** Let a *bit vector* be any infinite sequence of zeros and ones with only finitely many ones. A bit vector can be represented by a finite sequence, by cutting off the infinite zero suffix. Consider the relational structure $\mathbb{V} = (V, 0, +)$, consisting of the set V of all bit vectors, together with a unary predicate $0(\_)$ that distinguishes the zero vector, and the ternary relation $\_ + \_ = \_$ that describes point-wise addition modulo 2. Automorphisms of $\mathbb{V}$ are precisely linear mappings, i.e., bijections f s.t. $f(0) = 0$ and $f(u + v) = f(u) + f(v)$. The orbit of a tuple $(v_1, \ldots, v_n) \in V^n$ is determined by its *addition type*, i.e., by the the set of all equalities of the form $v_{i_1} + \ldots + v_{i_m} = 0$ satisfied by $(v_1, \ldots, v_n)$. Indeed, for two tuples $(u_1, \ldots, u_n), (v_1, \ldots, v_n) \in V^n$ having the same addition type, consider the partial bijection f defined as $f(u_1) = v_1, \ldots, f(u_n) = v_n$. By using the Steinitz exchange lemma, the function f can be extended to a linear mapping on the whole V, and thus $(u_1, \ldots, u_n)$ and $(v_1, \ldots, v_n)$ are in the same orbit. Therefore, the number of orbits of V^n is finite. On the other hand, $\mathbb{V}$ is not homogeneous. For instance, the two induced substructures $X = \{1000, 0100, 0010, 0001\}$ and $Y = \{1000, 0100, 0010, 1110\}$ are isomorphic. Define, e.g., $f(0001) = 1110$, and $f(x) = x$ if $x \neq 0001$. The reason why f is an isomorphism is that f needs to respect $+$ only in its domain, and any combination of two vectors from X falls outside of X. However, the isomorphism f does not extend to an automorphism of $\mathbb{V}$, since vectors in Y are not independent[4].

[4] The notion of homogeneity can be extended to structures with relations and functions, but one must consider *finitely-generated* induced substructures of $\mathbb{A}$ instead of finite ones. Note that $\mathbb{V}$ becomes homogeneous if $+$ is considered as a binary *function*, instead of a relation. The reason is that, in the presence of the functional symbol $+$, the homogeneity condition for $\mathbb{V}$ quantifies over finite induced substructures that are closed w.r.t. $+$, unlike the substructures in our example.

Fix a homogeneous relational structure $\mathbb{A}$. We give a precise complexity upper-bound for the complexity of the concrete saturation procedure from Fig. 2 and, thus, for reachability. This depends on the complexity of the induced substructure problem for $\mathbb{A}$. The *induced substructure problem* for $\mathbb{A}$ asks whether a given finite structure A over the same vocabulary is an induced substructure of $\mathbb{A}$. Assume that the induced substructure problem for $\mathbb{A}$ is decidable in time $T(k)$, where k is the size of the input. The complexity estimations below are always understood with respect to the sizes of the representing formulas. Let the *width* of a formula be the number of its variables. Let n be the width of an input automaton, defined as the greatest width of the formulas appearing in its definition, and let m be its *size*, defined as the sum of sizes of the defining formulas. By *T-relative pseudo-polynomial* time complexity we mean the time complexity

$$2^{\mathrm{poly}(n)} \cdot \mathrm{poly}(m) \cdot T(\mathrm{poly}(n)),$$

i.e., exponential in the width n but polynomial in the size m. Note that this is *relative* to the complexity T of the induced substructure problem.

▶ **Theorem 7.** *Let $\mathbb{A}$ be a homogeneous structure with induced substructure problem decidable in time $T(k)$. For FO-definable PDS $\mathcal{P}$ over $\mathbb{A}$ and a FO-definable NFA $\mathcal{A}$ recognizing a regular set of configurations $L_{\mathcal{P}}(\mathcal{A})$, one can construct in T-relative pseudo-polynomial time a FO-definable NFA $\mathcal{B}$ recognizing $L_{\mathcal{P}}(\mathcal{B}) = Reach_{\mathcal{P}}^{-1}(L_{\mathcal{P}}(\mathcal{A}))$.*

As a consequence, reachability in FO-definable PDS over $\mathbb{A}$ is decidable in T-relative pseudo-polynomial time.

Proof. Fix a homogeneous relational structure $\mathbb{A}$, and suppose that its induced substructure problem is decidable in time $T(k)$. We show that the concrete saturation algorithm from Fig. 2 terminates in T-relative pseudo-polynomial time. We use quantifier-free formulas over the vocabulary of $\mathbb{A}$ in *legal disjunctive normal form*, to be defined below. A *positive literal* is a predicate of the form $r(x_1, \ldots, x_k)$, where $x_1, \ldots, x_k$ are variables, and r is a relational symbol in the vocabulary of $\mathbb{A}$. A *negative literal* is the negation $\neg r(x_1, \ldots, x_k)$ of a positive literal, and a *literal* is either a positive or a negative literal. We treat equality in the same way as other relations of $\mathbb{A}$, thus there are also equality and inequality literals. A *clause* is a conjunction of pairwise different literals. A clause φ is *complete* if, for every positive literal l over the variables of φ, either l or its negation appears in φ, but not both. A complete clause φ is *consistent* if

- the equality literals define an equivalence over the variables of φ, and
- the literals of φ are invariant under this equivalence relation, i.e., replacing variables appearing in a literal of φ with equivalent ones yields a literal that also appears in φ.

A consistent clause φ gives rise to a finite structure $\mathcal{A}_\varphi$ over the same vocabulary as $\mathbb{A}$, whose elements are equivalence classes of variables, and where a relation $r([x_1], \ldots, [x_k])$ holds if, and only if, $r(x_1, \ldots, x_k)$ appears in φ (the choice of representative variables is irrelevant since φ is consistent). Thus, valuations satisfying φ are in one-to-one correspondence with *embeddings* of $\mathcal{A}_\varphi$ into $\mathbb{A}$, by which we mean injective homomorphisms that both preserve and reflect relations. A consistent clause φ is *legal* if, and only if, the structure $\mathcal{A}_\varphi$ is isomorphic to an induced substructure of $\mathbb{A}$, i.e., if there exists an embedding of $\mathcal{A}_\varphi$ into $\mathbb{A}$, written $\mathcal{A}_\varphi \sqsubseteq \mathbb{A}$. Thus, a clause φ is legal if, and only if, it is satisfiable.

▶ Proposition 2. Legality of a complete clause of size m is decidable in time $\mathrm{poly}(m) + T(m)$.

We consider two clauses to be equal when they contain the same literals. A formula is in *legal disjunctive normal form (ldnf)* if it is a disjunction of pairwise different legal clauses over the

same variables. We use the convention that the empty clause and the empty ldnf represent, respectively, true and false. For two formulas φ and ψ with the same free variables, we say that they are *equivalent*, written $\varphi \equiv \psi$, when $[\varphi] = [\psi]$, i.e., when they define the same set of tuples.

▶ Proposition 3. A quantifier-free formula φ can be transformed into an equivalent formula ψ in ldnf in T-relative pseudo-polynomial time.

Proof. Enumerate exhaustively all complete clauses over the variables of φ, and keep only those clauses $\{\psi_i\}_i$ which are legal (which is efficiently checkable by Proposition 2), and that satisfy φ (computable in time polynomial in the size of φ). Take $\psi = \bigvee_i \psi_i$. Clearly, $\psi \equiv \varphi$. The time complexity claim follows since the number of complete clauses is exponential in the number of variables, but independent from the size of φ. ◀

For homogeneous structures, the previous claim can be strengthened to first-order formulas. Essentially, this follows from the fact that, in a homogeneous structure, existential quantification can always be resolved positively.

▶ Proposition 4. A first-order formula φ can be transformed to an equivalent formula ψ in ldnf in T-relative pseudo-polynomial time.

Proof. As the first step, transform the input formula into prenex normal form. Then, transform the quantifier-free subformula into an equivalent ldnf, using Proposition 3. Finally, eliminate the quantifiers in sequence, starting from the innermost one, keeping the quantifier-free subformula in ldnf. Elimination of one existential quantifier is done as follows. First, distribute it over the disjunction of clauses,

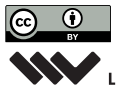
$$\varphi \quad \equiv \quad \exists x \cdot \psi_1 \vee \ldots \vee \psi_n \quad \equiv \quad \exists x \cdot \psi_1 \ \vee \ \ldots \ \vee \ \exists x \cdot \psi_n$$

and then replace every disjunct $\exists x \cdot \psi_i$ with the clause ψ_i' obtained from ψ_i by removing those literals that contain x. We claim that, after elimination of duplicates,

$$\varphi \quad \equiv \quad \psi_1' \ \vee \ \ldots \ \vee \ \psi_{n'}' \ ,$$

where the right-hand side is in lndf. To this end, we show that each ψ_i' is legal, and that $\exists x \cdot \psi_i \equiv \psi_i'$. Let $\mathcal{A}_{\psi_i}$ and $\mathcal{A}_{\psi_i'}$ be the two substructures of $\mathbb{A}$ defined by the two clauses. Clearly, $\mathcal{A}_{\psi_i'} \sqsubseteq \mathcal{A}_{\psi_i} \sqsubseteq \mathbb{A}$, which immediately implies legality of ψ_i' by transitivity. The left-to-right inclusion $[\exists x \cdot \psi_i] \subseteq [\psi_i']$ of the equivalence between $\exists x \cdot \psi_i$ and ψ_i' is immediate, since $\exists x \cdot \psi_i$ is more discriminating. For the other inclusion $[\psi_i'] \subseteq [\exists x \cdot \psi_i]$, let $\bar{a}' \in [\psi_i']$. Let $f_{\bar{a}'}$ be the natural embedding of $\mathcal{A}_{\psi_i'}$ into $\mathbb{A}$ mapping each equivalence class of variables in $\mathcal{A}_{\psi_i'}$ to the corresponding element in $\bar{a}'$. Similarly, since $\mathcal{A}_{\psi_i} \sqsubseteq \mathbb{A}$, there exists a tuple $\bar{a}b$ and an embedding $g_{\bar{a}b}$ of $\mathcal{A}_{\psi_i}$ into $\mathbb{A}$, where $g_{\bar{a}b}([x]) = b$. The substructure induced by $\bar{a}$ is isomorphic to that induced by $\bar{a}'$. Let h be such an isomorphism. Since $\mathbb{A}$ is homogeneous, h extends to a full automorphism of $\mathbb{A}$. Define $b' = h(b)$. Then, $\bar{a}'b' \in [\psi_i]$, and thus $\bar{a}' \in [\exists x \cdot \psi_i]$.

The universal quantifier is handled with the equivalence $\forall x \cdot \varphi \equiv \neg\exists x \cdot \neg\varphi$: First we replace $\neg\varphi$ by an equivalent formula in ldnf ψ by applying Proposition 3. Then, we apply the procedure above to remove the existential quantifier in $\exists x \cdot \psi$, and we thus obtain another formula ψ' in ldnf s.t. $\exists x \cdot \neg\varphi \equiv \psi'$. Finally, a further application of Proposition 3 to $\neg\psi'$ yields a formula ψ'' in ldnf s.t. $\psi'' \equiv \neg\exists x \cdot \neg\varphi$. ◀



By repeatedly using Proposition 4, we can implement the saturation algorithm in T-relative pseudo-polynomial time: First, transform all the formulas defining states and transitions of the input automata $\mathcal{P}$ and $\mathcal{A}$ into ldnf. Then, in every iteration, the formula forced(δ') is also transformed into ldnf. Step (2) is implemented by computing the union of clauses, and the implication in step (3) reduces to the inclusion of the sets of clauses of forced(δ') into those of δ'. Thus, one iteration of the algorithm requires relative pseudo-polynomial time. The total number of iterations is bounded by the number of orbits of the set $Q \times S \times Q$, since in every iteration at least one orbit is added to δ'. By Proposition 1, the number of orbits in bounded by $2^{\text{poly}(n)}$ where n is the dimension of $Q \times S \times Q$. Therefore, the concrete saturation algorithm runs in T-relative pseudo-polynomial time for homogeneous atoms. ◀

As a consequence of Theorem 7, under a bound on the width of input automata, the PDS reachability problem is in `PTime`, independently of the complexity $T(k)$ of the induced substructure problem. Moreover, the proof of Theorem 7 reveals that the polynomial above does not depend on the bound on width[5].

▶ **Corollary 8.** *The PDS reachability problem is fixed-parameter* `PTime`*, with the width of the input automaton as the parameter.*

In Theorem 7 we have shown that the complexity of the saturation procedure/reachability can be upper-bounded once we have a bound on the complexity of the induced substructure problem. We show below that, depending on the homogeneous structure, the latter problem (and thus reachability) can be of arbitrarily high complexity, or even undecidable. Therefore, the bound on the time complexity of induced substructure problem in Theorem 7 is a necessary assumption.

▶ **Theorem 9.** *Let $X \subseteq \mathbb{N}$ be a set of natural numbers. There exists a homogeneous structure $\mathbb{A}_X$ s.t. membership in X is many-one reducible to the induced substructure problem for $\mathbb{A}_X$.*

Proof. Let $X \subseteq \mathbb{N}$ be an arbitrary set of natural numbers. Intuitively, we effectively encode the set of natural numbers in an infinite antichain of finite tournaments, and we construct a homogeneous structure $\mathbb{A}_X$ s.t., for every natural number $n \in \mathbb{N}$, $n \in X$ if, and only if, the encoding of n is an induced substructure of $\mathbb{A}_X$. We use the instantiation of the embedding partial order $\sqsubseteq$ to finite directed graphs: $G \sqsubseteq H$ if G is isomorphic to an induced subgraph of H. A *tournament* is a directed graph $T = (V, E)$ s.t., for every pair of vertices $x, y \in V$, either $(x, y) \in E$, or $(y, x) \in E$, but not both. It is known that there exists a countably infinite $\sqsubseteq$-antichain $\mathcal{T}$ of finite tournaments [20]. Let f be an efficiently computable bijective mapping between natural numbers and tournaments in the antichain $\mathcal{T}$. Let $\mathcal{T}_X$ be those finite tournaments T in $\mathcal{T}$ with $T = f(n)$ for some $n \in X$. The construction of $\mathbb{A}_X$ uses the following result.

▶ Proposition 5 ([23]; see also [20]). For every $\sqsubseteq$-upward-closed family $\mathcal{T}$ of finite tournaments, there is a homogeneous directed graph $\mathbb{A}$ such that, for every finite tournament T, $T \sqsubseteq \mathbb{A}$ if, and only if, $T \in \mathcal{T}$.

Let $\mathbb{A}_X$ be the homogeneous directed graph obtained by applying the proposition above to the upward closure of the antichain $\mathcal{T}_X$. Then, for a natural number $n \in \mathbb{N}$, we have $n \in X$ if, and only if, the finite tournament $f(n)$ is in $\mathcal{T}_X$, which is the same as $f(n)$ being in the upward-closure of $\mathcal{T}_X$, since $f(n)$ is by construction in the antichain $\mathcal{T}$. By the proposition

[5] We are grateful to Mikołaj Bojańczyk for noticing this fact.

above, the latter property is equivalent to ask whether $f(n) \sqsubseteq \mathbb{A}_X$. Therefore, we can reduce membership in X to the induced substructure problem in $\mathbb{A}_X$. ◀

6 Examples of homogeneous structures

The purpose of this section is to provide concrete examples of homogeneous structures for which we can efficiently solve the reachability problem of FO-definable PDS. Those are well known in the model-theoretic community (cf. [23]), and we present them here in order to show the wide applicability of our results. We also present a general technique, called *wreath product*, which can be used to derive new homogeneous structures from known ones. Recall that, by Theorem 7, if $T(k)$ is the time complexity of the induced substructure problem of a homogeneous structure $\mathbb{A}$, then reachability of FO-definable PDS over $\mathbb{A}$ is decidable in T-relative pseudo-polynomial time. When the former problem is in `PTime`, reachability can be solved in `ExpTime` by the following corollary of Theorem 7.

▶ **Corollary 10.** *Let $\mathbb{A}$ be a homogeneous relational structure with a* `PTime` *induced substructure problem. For FO-definable PDS $\mathcal{P}$ over $\mathbb{A}$ and a FO-definable NFA $\mathcal{A}$ recognizing a regular set of configurations $L_{\mathcal{P}}(\mathcal{A})$, one can construct in* `ExpTime` *a FO-definable NFA $\mathcal{B}$ recognizing $L_{\mathcal{P}}(\mathcal{B}) = Reach_{\mathcal{P}}^{-1}(L_{\mathcal{P}}(\mathcal{A}))$. In particular, the FO-definable PDS reachability problem over $\mathbb{A}$ is in* `ExpTime`.

All the concrete examples that we provide in the sequel, and all infinitely many examples that can be obtained by applying the wreath product, have a `PTime` induced substructure problem, and thus reachability is in `ExpTime`.

Equality. Equality atoms $(\mathbb{D}, =)$ consist of a countably-infinite set $\mathbb{D}$ together with the equality relation. Automorphisms are permutations of $\mathbb{D}$. Homogeneity follows from the fact that any finite partial bijection $\mathbb{D} \to \mathbb{D}$ can be extended to a permutation of the whole set $\mathbb{D}$. This is arguably the simplest homogeneous structure. The induced substructure problem is in `PTime`, since it amounts to check whether the interpretation of $=$ in a given finite structure is the equality relation. By Corollary 10, reachability for FO-definable PDS over equality atoms is in `ExpTime`. This subsumes the result of [25], which considers a special case of our model where, among other restrictions, the input and stack alphabets are 1-dimensional, and the transition relation is quantifier-free definable (instead of FO-definable). Additionally, [25] shows that the problem is `ExpTime`-hard for equality atoms.

All the examples below generalize equality atoms by adding more relations to the vocabulary. We omit equality, which is assumed to always be in the vocabulary.

Equivalence. *Equivalence atoms* $(\mathbb{D}, R)$ consist of a countably-infinite set $\mathbb{D}$ and an equivalence relation R over $\mathbb{D}$ having infinitely-many, infinite equivalence classes. An automorphism of equivalence atoms is a bijection f of $\mathbb{D}$ which respects R, in the sense that, for every $x, y \in \mathbb{D}$, $(x, y) \in R$ if, and only if, $(f(x), f(y)) \in R$. Equivalence atoms are homogeneous. (We will see later that equivalence atoms are isomorphic with the wreath product of equality atoms with itself.) This can model hierarchically nested data, where one can check whether two elements belong to the same equivalence class, and, if so, whether they actually are the same element. Higher nested equivalence atoms can be obtained by iterating this process: 0-nested equivalence atoms are just equality atoms; and for any $k \geq 0$, $(k+1)$-nested equivalence atoms can be seen as the disjoint union of infinitely many copies of k-nested equivalence atoms, with one additional equivalence relation that relates a pair of elements if, and only if, they belong to the same copy.

Total, betweenness, and cyclic order. *Total order atoms* $(\mathbb{Q}, \leq)$ can be presented as the rational numbers $\mathbb{Q}$ together with the natural total order $\leq$. Automorphisms are monotonic bijections of rational numbers. Homogeneity follows from the fact that $\leq$ is dense: A monotonic bijection $f : X \to Y$ over a finite domain X extends to an automorphism of $\mathbb{Q}$. The induced substructure problem is in `PTime`, since it amounts to check whether the interpretation of $\leq$ in a given finite structure is a total order. This can be used to model qualitative time, where events are totally ordered, but no information is available on the distance between them. Another instance is given by data-centric applications [15].

Betweenness order atoms $(\mathbb{Q}, B)$ use the betweenness relation B, which is obtained by considering the order $\leq$ up to reversal: $B(x, y, z)$ holds when x lies between y and z, i.e., either $y < x < z$ or $z < x < y$. This can be used to model time where one is not interested on the order between the events themselves, but rather on whether an event happened between two other events. *Cyclic order atoms* $(\mathbb{Q}, K)$ use the ternary cyclic ordering K obtained by bending the total order into a circle. Formally, $K(x, y, z)$ if either $x < y < z$, or $z < x < y$, or $y < z < x$. This can model a notion of qualitative cyclic time, where events cyclically repeat, but no precise timing information is available. For both betweenness and cyclic order atoms, the induced substructure problem is in `PTime`.

Universal partial order and preorder. Every relational homogeneous structure is obtained as the *Fraïssé limit* of the set of all its finite induced substructures [18]. (We do not formally define here the notion of Fraïssé limit, which is a central tool for constructing homogeneous structures; cf. [23].) For instance, total order atoms are the Fraïssé limit of all finite total orders. *Partial order atoms* are obtained as the Fraïssé limit of the set of all finite partial orders. The induced substructure problem amounts to determine whether the interpretation of $\leq$ in a given finite structure is a partial order, which can clearly be done in `PTime`. This can be used to model the ordering of events in distributed systems. Along the same lines one obtains *preorder atoms*.

Universal tree order. A *tree order* (or semilinear order) is a partially ordered structure $(A, \leq)$ s.t. a) every two elements have an common upper bound, and b) for every element, its upward closure is totally ordered. Tree order atoms $(T, \leq)$ are obtained as the Fraïssé limit of the set of all finite tree orders. Intuitively, tree order atoms consists of a countably-infinite tree order where each maximal path is isomorphic to total order atoms. Unfortunately, tree order atoms as presented here are not homogeneous. Intuitively, this happens because isomorphic substructures have least upper bounds outside the structures themselves, and they might relate to those in an incomparable way. This can be amended by introducing be the following ternary relation: $R(x, y, z)$ holds when the lub of x and y is incomparable with z. Then, $(T, \leq, R)$ is homogeneous, and it can be obtained as the Fraïssé limit of the set of all extended finite tree orders $(A, \leq, R)$. Clearly, the induced substructure problem is in `PTime` for $(T, \leq, R)$.

Universal graph and tournament. *Universal graph atoms* are obtained as the Fraïssé limit of the set of all finite graphs. This is also known as *Rado's graph* or *the random graph*. The induced substructure problem is trivial since the universal graph contains an isomorphic copy of every finite graph. Similarly, *universal tournament atoms* are the Fraïssé limit of the set of all finite tournaments, where a tournament is an irreflexive graph $T = (V, E)$ s.t., for every two nodes $x, y \in V$, either $(x, y) \in E$, or $(y, x) \in E$. Given a graph, it is clearly checkable in `PTime` whether it is actually a tournament, thus the induced substructure problem is in

PTime also in this case.

Wreath products. We conclude this section by giving a construction which allows to compose homogeneous structures in order to produce new ones. Given two relational structures $\mathbb{A} = (A, R_1, \ldots, R_m)$ and $\mathbb{B} = (B, S_1, \ldots, S_n)$, their *wreath product* is the relational structure $\mathbb{A} \otimes \mathbb{B} = (A \times B, R'_1, \ldots, R'_m, S'_1, \ldots, S'_n)$, where $((a_1, b_1), \ldots, (a_k, b_k)) \in R'_i$ if $(a_1, \ldots, a_k) \in R_i$, and $((a_1, b_1), \ldots, (a_k, b_k)) \in S'_j$ if $a_1 = \cdots = a_k$ and $(b_1, \ldots, b_k) \in S_j$. Intuitively, $\mathbb{A} \otimes \mathbb{B}$ is obtained by replacing each element in $\mathbb{A}$ with a disjoint copy of $\mathbb{B}$. It can be checked that, if the two structures $\mathbb{A}$ and $\mathbb{B}$ are homogeneous, then the same holds for their wreath product $\mathbb{A} \otimes \mathbb{B}$. The induced substructure problem for $\mathbb{A} \otimes \mathbb{B}$ reduces in PTime to the same problem for $\mathbb{A}$ and $\mathbb{B}$: $\{(a_1, b_1), \ldots, (a_k, b_k)\}$ is an induced substructure of $\mathbb{A} \otimes \mathbb{B}$ if, and only if, $\{a_1, \ldots, a_k\}$ is an induced substructure of $\mathbb{A}$, and for every i, $\{b_j \mid a_j = a_i\}$ is an induced substructure of $\mathbb{B}$. Therefore, if both $\mathbb{A}$ and $\mathbb{B}$ have a PTime induced substructure problem, then the same holds for $\mathbb{A} \otimes \mathbb{B}$, and Corollary 10 applies.

As an application of the wreath product, take $\mathbb{A}_0 = (\mathbb{D}, =)$ to be equality atoms, and, for each $k \geq 0$, let $\mathbb{A}_{k+1} = \mathbb{A}_0 \otimes \mathbb{A}_k$. Then, $\mathbb{A}_1$ is just the equivalence atoms presented before, and, more generally, $\mathbb{A}_k = (\mathbb{D}, R_1, \ldots, R_k)$ is *k-nested equivalence atoms*, which can be used to model data with nested equivalence relations. For each of those infinitely many examples, the reachability problem for FO-definable PDS is in ExpTime.

7 Conclusions

We have studied the reachability problem for a model of PDS with countably-infinite FO-definable states, stack alphabet, and transitions relation. We advocate a Ockham's razor research strategy that refrains from inventing seemingly new notions. Instead, we have taken the standard definition of PDS and re-interpreted it in the richer framework of FO-definable sets instead of ordinary finite sets. This covers the well-known model of pushdown register automata [11, 25] as one instantiation of the general paradigm, and we have shown that the optimal ExpTime complexity for the reachability problem for this model can be recovered in the more general framework. This same paradigm can of course be applied to a variety of different models, like timed PDS [2], data/timed extensions of Petri nets [3, 22], lossy channel systems [1], 1-clock/1-register alternating automata [21, 26, 14], rewriting systems [7], etc. Therefore, the present paper can be seen as a proof of concept of the new research strategy. For example, one could consider *FO-definable pushdown automata* (PDA) and *FO-definable context-free grammars* (CFG) as acceptors of languages over infinite alphabets. The definition of FO-definable PDA is analogous to PDS, except that the transition relation is a FO-definable subset of $Q \times \Gamma^* \times A_\varepsilon \times Q \times \Gamma^*$, where $A_\varepsilon = A \cup \{\varepsilon\}$ is an FO-definable alphabet extended with the empty word. Similarly, FO-definable CFG can be defined as *stateless* FO-definable PDA where every transition pops exactly one symbol from the stack. It is easy to prove that FO-definable PDA languages coincide with FO-definable context-free languages for oligomorphic atoms [5], and that the latter are closed under union, concatenation, Kleene star, homomorphism, inverse homomorphism, intersection with FO-definable regular languages, and that collapsing each orbit to a different symbol yields a classical context-free language.

References

1 P. A. Abdulla, M. F. Atig, and J. Cederberg. Timed lossy channel systems. In *Proc. of FSTTCS'12*, volume 18 of *LIPIcs*, pages 374–386, 2012.

2 P. A. Abdulla, M. F. Atig, and J. Stenman. Dense-timed pushdown automata. In *Proc. of LICS'12*, pages 35–44, june 2012.

3 P. A. Abdulla and A. Nylén. Timed Petri nets and BQOs. In *Proc. of ICATPN'01*, pages 53–70, 2001.

4 M. F. Atig. Model-checking of ordered multi-pushdown automata. *Log. Methods Comput. Sci.*, 8(3), 09 2012.

5 M. Bojańczyk, B. Klin, and S. Lasota. Automata theory in nominal sets. *Logical Methods in Computer Science*, 10(3:4):paper 4, 2014.

6 A. Bouajjani, J. Esparza, and O. Maler. Reachability analysis of pushdown automata: Application to model-checking and saturation method. In *Proc. of CONCUR'97*, volume 1243 of *LNCS*, pages 135–150, 1997.

7 A. Bouajjani, P. Habermehl, Y. Jurski, and M. Sighireanu. Rewriting systems with data. In *In Proc. of FCT'07*, volume 4639 of *LNCS*, pages 1–22. Springer, 2007.

8 L. Breveglieri, A. Cherubini, C. Citrini, and S. Crespi-Reghizzi. Multi-push-down languages and grammars. *Int. J. Found. Comput. Sci.*, 7(3):253–292, 1996.

9 C. Broadbent, A. Carayol, M. Hague, and O. Serre. A saturation method for collapsible pushdown systems. In *Proc. of ICALP'12*, volume 7392 of *LNCS*, pages 165–176, 2012.

10 A. Carayol and M. Hague. Saturation algorithms for model-checking pushdown systems. In *Proc. of AFL'14*, volume 151 of *EPTCS*, pages 1–24, 5 2014.

11 E. Y. C. Cheng and M. Kaminski. Context-free languages over infinite alphabets. *Acta Inf.*, 35(3):245–267, 1998.

12 L. Clemente and S. Lasota. Timed pushdown automata revisited. In *Proc. of LICS'15*, 2015. Accepted for publication.

13 W. Czerwiński, P. Hofman, and S. Lasota. Reachability problem for weak multi-pushdown automata. *Logical Methods in Computer Science*, 9(3:13):1–29, 2013.

14 S. Demri and R. Lazic. LTL with the freeze quantifier and register automata. *ACM Trans. Comput. Logic*, 10(3):16:1–16:30, Apr. 2009.

15 A. Deutsch, R. Hull, F. Patrizi, and V. Vianu. Automatic verification of data-centric business processes. In *Proc. of ICDT'09*, pages 252–267, New York, NY, USA, 2009. ACM.

16 J. Esparza and S. Schwoon. A BDD-based model checker for recursive programs. In *Proc. of CAV'01*, CAV '01, pages 324–336. Springer-Verlag, 2001.

17 A. Finkel, B. Willems, and P. Wolper. A direct symbolic approach to model checking pushdown systems. In *Proc. of INFINITY'97*, volume 9, pages 27–37, 1997.

18 R. Fraïssé. *Theory of relations*. North-Holland, 1953.

19 I. Guessarian. Pushdown tree automata. *Theor. Comp. Sys.*, 16:237–263, 1983.

20 W. Henson. Countable homogeneous relational structures and $\aleph_0$-categorical theories. *J. Symb. Logic*, 37:494–500, 1972.

21 S. Lasota and I. Walukiewicz. Alternating timed automata. *ACM Trans. Comput. Logic*, 9(2):10:1–10:27, 2008.

22 R. Lazic, T. Newcomb, J. Ouaknine, A. W. Roscoe, and J. Worrell. Nets with tokens which carry data. In *Proc. of ICATPN'07*, LNCS, pages 301–320. Springer-Verlag, 2007.

23 D. Macpherson. A survey of homogeneous structures. *Discrete Mathematics*, 311(15):1599–1634, 2011.

24 A. N. Maslov. Multilevel stack automata. *Probl. Peredachi Inf.*, 12(1):55–62, 1976.

25 A. S. Murawski, S. J. Ramsay, and N. Tzevelekos. Reachability in pushdown register automata. In *MFCS 2014*, pages 464–473, 2014.

26 J. Ouaknine and J. Worrell. On the language inclusion problem for timed automata: Closing a decidability gap. In *In Proc. of LICS'04*, pages 54–63. IEEE Computer Society, 2004.