Well-Structured Pushdown Systems

Xiaojuan Cai[1] and Mizuhito Ogawa[2]

[1] BASICS Lab, Shanghai Jiao Tong University, China
cxj@sjtu.edu.cn
[2] Japan Advanced Institute of Science and Technology, Japan
mizuhito@jaist.ac.jp

Abstract. *Pushdown systems* (PDSs) model single-thread recursive programs, and *well-structured transition systems* (WSTSs), such as *vector addition systems*, are useful to represent non-recursive multi-thread programs. Combining these two ideas, our goal is to investigate *well-structured pushdown systems* (WSPDSs), pushdown systems with well-quasi-ordered control states and stack alphabet.

This paper focuses on subclasses of WSPDSs, in which the coverability becomes decidable. We apply WSTS-like techniques on classical P-automata. A *Post**-automata (resp. *Pre**-automata) construction is combined with Karp-Miller acceleration (resp. ideal representation) to characterize the set of successors (resp. predecessors) of given configurations. As examples, we show that the coverability is decidable for *recursive vector addition system with states*, *multi-set pushdown systems*, and a WSPDS with finite control states and well-quasi-ordered stack alphabet.

1 Introduction

There are two directions of infinite (discrete) state systems. A *pushdown system* (PDS) consists of finite control states and finite stack alphabet, where a stack stores the context. It is often used to models single-thread recursive programs.A *well-structured transition system* (WSTS) [1,10] consists of a well-quasi-ordered set of states. A *vector addition system* (VAS, or Petri Net) is its typical example. It often works for modeling dynamic thread creation of multi-thread program [2]. Our naive motivation comes from what happens when we combine them as a general framework for modeling recursive multi-thread programs.

A 3-thread boolean-valued recursive program with synchronization is enough to encode *Post-correspondence-problem* [19]. Thus, its reachability is undecidable. There are several decidable subclasses, which are typically reduced to single stack PDSs with infinite control states and stack alphabet.

- *Restrict the number of context switching (bounded reachability):* Context-bounded concurrent pushdown systems [18], and their extensions with dynamic thread creation [2].
- *Restrict interleaves among stack operations:* Multi-set pushdown systems (Multi-set PDSs) to model multi-thread asynchronous programs [20,13], and Recursive Vector Addition System with States (RVASS) to model multi-thread programs with fork/join synchronizations [3].

P.R. D'Argenio and H. Melgratti (Eds.): CONCUR 2013, LNCS 8052, pp. 121–136, 2013.

A popular decidable property of ordinary PDSs is the *configuration reachability*, i.e., whether a target configuration is reachable from an initial configuration. A P-automaton construction [9,4,7] is its classical technique such that a $Post^*$ automaton accepts the set of successors of an initial configuration, and a Pre^* automaton accepts the set of predecessors of a target configuration.

A popular decidable property of WSTSs is *coverability*, i.e., whether an initial configuration reaches to that covers a target configuration. There are forward and backward techniques. As the former, Karp-Miller acceleration [8] for VASs is well-known, which was generalized in [11,12]. As the latter, an ideal (i.e., an upward closed set) representation is immediate [1,10], though less efficient. Note that the reachability of WSTSs is not easy. For instance, the reachability of VASs stays decidable, but it requires deep insight on Presburger arithmetic [16,15].

Our ultimate goal is to study *well-structured pushdown systems* (WSPDSs), pushdown systems with well-quasi-ordered control states and stack alphabet. This paper focuses on subclasses of WSPDSs, in which the coverability becomes decidable. We apply WSTS-like techniques on classical P-automata. A $Post^*$-automata (resp. Pre^*-automata) construction is combined with Karp-Miller acceleration (resp. ideal representation) to characterize the set of successors (resp. predecessors) of given configurations. As examples, we show that the coverability is decidable for RVASSs, Multi-set PDSs, and a WSPDS with finite control states and WQO stack alphabet. The first one extends the decidability of the state reachability of RVASSs [3] to the coverability, and the second one relaxes finite stack alphabet of Multi-set PDSs [20,13] to being well-quasi-ordered.

Related Work

Combining PDSs and VASs is not new. Process rewrite system (PRS) [17] is a pioneer work on such combination. A PRS is a(n AC) ground term rewriting system, consisting of the sequential composition ".", the parallel composition "|| ", and finitely many constants, which can be regarded as a PDS with finite control states and vector stack alphabet. The decidability of the reachability between ground terms was shown based on the reachability of a VAS. However, a PRS is rather weak to model multi-thread programs, since it cannot describe vector additions between adjacent stack frames during push/pop operations.

An RVASS [3] allows vector additions during pop rules. The state reachability was shown by reducing an RVASS to a Branching VASS [21]. Our WSPDS extends it to the coverability. A more general framework is a WQO automaton [5], which is a WSTS with auxiliary storage (e.g., stacks and queues). Although in general undecidable, its coverability becomes decidable under the compatibility of *rank* functions with a WQO. A Multi-set PDS [13,20] is a such instance.

Our drawback is difficulty to estimate complexity, due to the nature of well-quasi-ordering. s For instance, the coverability of a Branching VAS (BVAS) is 2EXPTIME-complete [6], and accordingly RVASS will be. Lower bounds of various VAS are reported by reduction to fragments of first-order logic [14]. However, we cannot directly conclude such estimations.

2 Preliminaries

2.1 Well-Structured Transition System

A *quasi-order* $(D, \leq)$ is a reflexive transitive binary relation on D. An upward closure of $X \subseteq D$, denoted by $X^{\uparrow}$, is the set of elements in D larger than those in X, i.e., $X^{\uparrow} = \{d \in D \mid \exists x \in X.x \leq d\})$. A subset I is an *ideal* if $I = I^{\uparrow}$. Similarly, a downward closure of $X \subseteq D$ is denoted by $X^{\downarrow} = \{d \in D \mid \exists x \in X.x \geq d\}$. We denote the set of all ideals by $\mathcal{I}(D)$. A quasi-order $(D, \leq)$ is a *well-quasi-order* (WQO) if, for each infinite sequence $a_1, a_2, a_3, \cdots$ in D, there exist i, j with $i < j$ and $a_i \leq a_j$.

Definition 1. *A* well-structured transition system *(WSTS) is a triplet $M = \langle (P, \preceq), \rightarrow \rangle$ where $(P, \preceq)$ is a WQO, and $\rightarrow$ $(\subseteq P \times P)$ is monotonic, i.e., for each $p_1, q_1, p_2 \in P$, $p_1 \rightarrow q_1$ and $p_1 \preceq p_2$ imply that there exists q_2 with $p_2 \rightarrow q_2 \ \wedge q_1 \preceq q_2$.*

Given two states $p, q \in P$, the *coverability* problem is to determine whether there exists q' with $q' \succeq q$ and $p \rightarrow^* q'$.

Vector addition systems (VAS) (equivalently, Petri net) are WSTSs with $\mathbb{N}^k$ as the set of states and a subtraction followed by an addition as a transition rule. The reachability problem of VAS is decidable, but its proof is complex [16,15]. The coverability also attracts attentions and is implemented, such as in **Pep**. [1] *Karp-Miller acceleration* is an efficient technique for the coverability. If there is a descendant vector (wrt transitions) strictly larger than one of its ancestors on coordinates, values at these coordinates are accelerated to ω.

There is an alternative backward method to decide coverability for a general WSTS. Starting from an ideal $\{q\}^{\uparrow}$, where q is the target state to be covered, its predecessors are repeatedly computed. Note that, for a WSTS and an ideal $I(\subseteq P)$, the predecessor set $pre(I) = \{p \in P \mid \exists q \in I.p \rightarrow q\}$ is also an ideal from the monotonicity. Its termination is obtained by the following lemma.

Lemma 1. *[10] $(D, \leq)$ is a WQO, if, and only if, any infinite sequence $I_0 \subseteq I_1 \subseteq I_2 \subseteq \cdots$ in $\mathcal{I}(D)$ eventually stabilize.*

From now on, we denote $\mathbb{N}$ (resp. $\mathbb{Z}$) for the set of natural numbers (resp. integers), and $\mathbb{N}^k$ (resp. $\mathbb{Z}^k$) is the set of k-dimensional vectors over $\mathbb{N}$ (resp. $\mathbb{Z}$). As notational convention, $\boldsymbol{n}, \boldsymbol{m}$ are for vectors in $\mathbb{N}^k$, $\boldsymbol{z}, \boldsymbol{z}'$ are for vectors in $\mathbb{Z}^k$, $\widetilde{\boldsymbol{n}}, \widetilde{\boldsymbol{m}}$ are for sequences of vectors.

2.2 Pushdown System

We define a pushdown system (PDS) with extra rules, *simple-push* and *nonstandard-pop*. These rules do not appear in the standard definition since they are encoded into standard rules. For example, a non-standard pop rule

[1] `http://theoretica.informatik.uni-oldenburg.de/p̃ep/`

$(p, \alpha\beta \rightarrow q, \gamma)$ is split into $(p, \alpha \rightarrow p_\alpha, \epsilon)$ and $(p_\alpha, \beta \rightarrow q, \gamma)$ by adding a fresh state p_α. However, later we will consider a PDS with infinite stack alphabet, and this encoding may change the context. For instance, for a PDS with finite control states and infinite stack alphabet, this encoding may lead infinite control states.

Definition 2. *A* pushdown system (PDS) *is a triplet* $\langle P, \Gamma, \Delta\rangle$ *where*

- P *is a finite set of states,*
- Γ *is finite stack alphabet, and*
- $\Delta \subseteq P \times \Gamma^{\leq 2} \times P \times \Gamma^{\leq 2}$ *is a finite set of transitions, where* $(p, v, q, w) \in \Delta$ *is denoted by* $(p, v \rightarrow q, w)$.

We use $\alpha, \beta, \gamma, \cdots$ to range over Γ, and $w, v, \cdots$ over words in Γ^*. A *configuration* $\langle p, w\rangle$ is a pair of a state p and a stack content (word) w. As convention, we denote configurations by $c_1, c_2, \cdots$. One step transition $\hookrightarrow$ between configurations is defined as follows. $\hookrightarrow^*$ is the reflexive transitive closure of $\hookrightarrow$.

$$\text{inter}\frac{(p,\gamma \rightarrow p', \gamma') \in \Delta}{\langle p, \gamma w\rangle \hookrightarrow \langle p', \gamma' w\rangle} \quad \text{push}\frac{(p,\gamma \rightarrow p', \alpha\beta) \in \Delta}{\langle p, \gamma w\rangle \hookrightarrow \langle p', \alpha\beta w\rangle} \quad \text{pop}\frac{(p,\gamma \rightarrow p', \epsilon) \in \Delta}{\langle p, \gamma w\rangle \hookrightarrow \langle p', w\rangle}$$

$$\text{simple-push}\frac{(p,\epsilon \rightarrow p', \alpha) \in \Delta}{\langle p, w\rangle \hookrightarrow \langle p', \alpha w\rangle} \quad \text{nonstandard-pop}\frac{(p,\alpha\beta \rightarrow p', \gamma) \in \Delta}{\langle p, \alpha\beta w\rangle \hookrightarrow \langle p', \gamma w\rangle}$$

A PDS enjoys decidable *configuration reachability*, i.e., given configurations $\langle p, w\rangle$, $\langle q, v\rangle$ with $p, q \in P$ and $w, v \in \Gamma^*$, decide whether $\langle p, w\rangle \hookrightarrow^* \langle q, v\rangle$.

3 WSPDS and P-Automata Technique

3.1 P-Automaton

A P-automaton is an automaton that accepts the set of reachable configurations of a PDS. P-automata are classified into *Post**-automata and *Pre**-automata,

Definition 3. *Given a PDS* $M = \langle P, \Gamma, \Delta\rangle$, *a P-automaton* $\mathcal{A}$ *is a quadruplet* (S, Γ, ∇, F) *where*

- F *is the set of final states, and* $P \subseteq S \setminus F$, *and*
- $\nabla \subseteq S \times (\Gamma \cup \{\epsilon\}) \times S$.

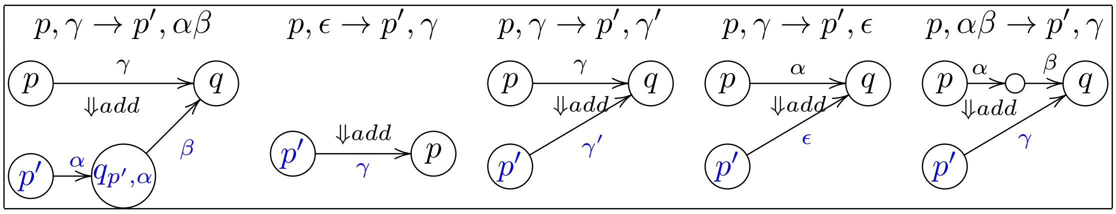
We write $s \overset{\gamma}{\mapsto} s'$ *for* $(s, \gamma, s') \in \nabla$ *and* $\Mapsto$ *for the reflexive transitive closure of* $\mapsto$*; It* accepts $\langle p, w\rangle$ *for* $p \in P$ *and* $w \in \Gamma^*$ *if* $p \overset{w}{\Longmapsto} f \in F$. *We use* $L(\mathcal{A})$ *to denote the set of configurations that* $\mathcal{A}$ *accepts. We assume that an* initial P-automaton *has no transitions* $s \overset{\gamma}{\mapsto} s'$ *with* $s' \in P$.

Let C_0 be a regular set of configurations of a PDS, and let $\mathcal{A}_0$ be an initial P-automaton that accepts C_0. The procedure to compute $post^*(C_0)$ starts from $\mathcal{A}_0$, and repeatedly adds edges according to the rules of a PDS until convergence. We call this procedure *saturation*. *Post**-saturation rules are given in Definition 4, which are illustrated in the following figure.

Definition 4. *For a PDS $\langle P, \Gamma, \Delta\rangle$, let $\mathcal{A}_0$ be an initial P-automaton accepting C_0. $Post^*(\mathcal{A}_0)$ is constructed by repeated applications of the following $Post^*$-saturation rules.*

$$\frac{(S, \Gamma, \nabla, F), \quad (p \overset{w}{\Longmapsto} q) \in \nabla}{(S \cup \{p'\}, \Gamma, \nabla \cup \{p' \overset{\gamma}{\mapsto} q\}, F)} \ (p, w \to p', \gamma) \in \Delta, |w| \leq 2$$

$$\frac{(S, \Gamma, \nabla, F), \quad (p \overset{\gamma}{\mapsto} q) \in \nabla}{(S \cup \{p', q_{p',\alpha}\}, \Gamma, \nabla \cup \{p' \overset{\alpha}{\mapsto} q_{p',\alpha} \overset{\beta}{\mapsto} q\}, F)} \ (p, \gamma \to p', \alpha\beta) \in \Delta$$



For instance, consider a push rule $(p, \gamma \to p', \alpha\beta)$. If $p \overset{\gamma}{\mapsto} q$ is in ∇, then $p' \overset{\alpha}{\mapsto} q_{p',\alpha} \overset{\beta}{\mapsto} q$ is added to ∇. The intuition is, if, for $v \in \Gamma^*$, $\langle p, \gamma v\rangle$ is in $post^*(C_0)$, then $\langle p', \alpha\beta v\rangle$ is also in $post^*(C_0)$ by applying rule $(p, \gamma \to p', \alpha\beta)$. The Pre^*-saturation rules to construct $pre^*(C_0)$ are similar, but in the reversal.

Remark 1. $Post^*$- (resp. Pre^*-) saturation introduces ϵ-transitions when applying standard pop rules (resp. simple push rules). ϵ-transitions make arguments complicated, and we assume preprocessing on PDSs.

1. The bottom symbol $\bot$ of the stack is explicitly prepared in Γ.
2. For $Post^*$-saturation, each standard pop rule $p, \alpha \to q, \epsilon$ is replaced with $(p, \alpha\gamma \to q, \gamma)$ for each $\gamma \in \Gamma$.
3. For Pre^*-saturation, each *simple push* rule $p, \epsilon \to q, \alpha$ is replaced with $(p, \gamma \to q, \alpha\gamma)$ for each $\gamma \in \Gamma$.

Lemma 2. *Let $\langle P, \Gamma, \Delta\rangle$ be a PDS, and let $\mathcal{A}_0$ be an initial P-automaton accepting C_0. Assume that $p \overset{w}{\Longmapsto} q$ in $Post^*(\mathcal{A}_0)$ and $p \in P$.*

1. *If $q \in P$, $\langle q, \epsilon\rangle \hookrightarrow^* \langle p, w\rangle$;*
2. *If $q \in S(\mathcal{A}_0)\backslash P$, there exists $q' \overset{v}{\Longmapsto} q$ in $\mathcal{A}_0$ with $q' \in P$ and $\langle q', v\rangle \hookrightarrow^* \langle p, w\rangle$.*

Its proof is a folklore (also in [23]). Lemma 2 shows that each accepted configuration is in $post^*(C_0)$ during the saturation process (*soundness*). On the other hand, $Post^*$ saturation rules put immediate successor configurations, and all configurations in $post^*(C_0)$ are finally accepted by $Post^*(\mathcal{A}_0)$ (*completeness*).

Theorem 1. *$post^*(C_0) = L(Post^*(\mathcal{A}_0))$, and $pre^*(C_0) = L(Pre^*(\mathcal{A}_0))$.*

For an ordinary PDS (i.e., with finite control states and stack alphabet), $Post^*(\mathcal{A}_0)$ and $Pre^*(\mathcal{A}_0)$ have bounded numbers of states. (Recall that each newly added state $q_{p,\gamma}$ has an index of a pair of a state and a stack symbol.)

Thus, the saturation procedure finitely converges. For a PDS with infinite control states and stack alphabet, although $Post^*(\mathcal{A}_0)$ and $Pre^*(\mathcal{A}_0)$ may not finitely converge, they converge as limits (of set unions). The same statement to Theorem 1 holds by Lemma 2' (a generalized Lemma 2) in [23]. In later sections (Section 4 and 5), we show when and how the finite convergence holds.

3.2 P-Automata for Coverability

We denote the set of partial functions from X to Y by $\mathcal{P}Fun(X,Y)$. Let $\lll$, the quasi-ordering[2] on Γ^*, be the element-wise extension of $\leq$ on Γ, i.e., $\alpha_1 \cdots \alpha_n \lll \beta_1 \cdots \beta_m$ if and only if $m = n$ and $\alpha_i \leq \beta_i$ for each i.

Definition 5. *A* well-structured pushdown system *(WSPDS) is a triplet* $M = \langle (P, \preceq), (\Gamma, \leq), \Delta \rangle$ *where*

- $(P, \preceq)$ *and* $(\Gamma, \leq)$ *are WQOs, and*
- $\Delta \subseteq \mathcal{P}Fun(P,P) \times \mathcal{P}Fun(\Gamma^{\leq 2}, \Gamma^{\leq 2})$ *is the finite set of monotonic transitions rules (wrt* $\preceq$ *and* $\lll$*). We denote* $(p, w \to \phi(p), \psi(w))$ *if* $(\phi, \psi) \in \Delta$, $p \in Dom(\phi)$, *and* $w \in Dome(\psi)$ *hold.*

A PDS is a WSPDS with finite P and finite Γ, and WSTS is a WSPDS with a single control state and internal transition rules only (i.e., no push/pop rules). Note that $Dom(\psi)$ and $Dome(\phi)$ are upward-closed sets from their monotonicity. Instead of reachability, we consider the *coverability* on WSPDSs.

- **Coverability**: Given configurations $\langle p, w \rangle$, $\langle q, v \rangle$ with $p, q \in P$ and $w, v \in \Gamma^*$, we say $\langle p, w \rangle$ *covers* $\langle q, v \rangle$ if there exist $q' \succeq q$ and $v' \ggg v$ s.t. $\langle p, w \rangle \hookrightarrow^* \langle q', v' \rangle$. Coverability problem is to decide whether $\langle p, w \rangle$ *covers* $\langle q, v \rangle$.

Remark 2. Thanks to an anonymous referee, the coverability of a WSPDS is reduced to the state reachability. Let $v = \alpha_n \cdots \alpha_1 \bot$ and $v' = \beta_n \cdots \beta_1 \bot$. For fresh states $q_n, \cdots, q_1, q_0$ (incomparable wrt $\preceq$), add transition rules

$$\{(q', x \to q_n, \epsilon) \text{ if } x \geq \alpha_n \text{ and } q' \succeq q, \quad (q_{i+1}, x \to q_i, \epsilon) \text{ if } x \geq \alpha_i, \quad (q_1, \bot \to q_0, \bot)\}.$$

Then, the coverability (from $\langle p, w \rangle$ to $\langle q, v \rangle$) is reduced to the state reachability (from $\langle p, w \rangle$ to q_0). Note that the same technique (replacing $\geq$ and $\succeq$ with $=$) does not work for the configuration reachability, since it violates the monotonicity. Nevertheless, we keep focusing on the coverability, since

- Transition rules above are not permitted as an RVASS and a Multi-set PDS. Thus, the coverability is still more than the state reachability at the level of RVASSs and Multi-set PDSs.
- Proofs are mostly by induction on the saturation steps of P-automata construction. The coverability fits for describing their inductive invariants.

[2] In general, $\lll$ is not a well-quasi-ordering, even if $\leq$ is.

There are two ways to decide the coverability. The forward method starts from an initial configuration $\langle p, w\rangle$, and computes the downward closure of its successor configurations. The backward method starts from a target configuration $\langle q, v\rangle$, and computes the downward closure of its predecessor configurations.

- **(Post)** $\mathcal{A}$ accepts the downward closure of successors of C_0, i.e., $L(\mathcal{A}) = \bigcup_{i\geq 0}(post^i(C_0)^{\downarrow}) = (\bigcup_{i\geq 0} post^i(C_0))^{\downarrow} = (post^*(C_0))^{\downarrow}$.
- **(Pre)** $\mathcal{A}$ accepts predecessors of the upward closure $C_0^{\uparrow}$ of C_0, i.e., $L(\mathcal{A}) = \bigcup_{i\geq 0} pre^i(C_0^{\uparrow}) = pre^*(C_0^{\uparrow})$.

Remark 3. As in Remark 1, we preprocess WSPDSs to eliminate standard pop rules for $Post^*$-saturation and simple push rules for Pre^*-saturation. In later decidability results on WSPDSs, the finiteness of transition rules is crucial. The following replacement keeps the monotonicity and the finiteness.
- In $Post^*$-saturation, a standard pop rule $\psi(\gamma) = \epsilon$ is replaced with $\psi'(\gamma\gamma') = \gamma'$.
- In Pre^*-saturation, a simple push rule $\psi(\epsilon) = \gamma$ is replaced with $\psi'(\gamma') = \gamma\gamma'$.

4 *Post**-automata for Coverability

Coverability is decidable if either $Post^*$ or Pre^*-saturation finitely converges. In this section, we consider a strictly monotonic WSPDS with finitely many control states, with $\mathbb{N}^k$ as stack alphabet, and without *standard push* rules. Such a PDS is a *Pushdown Vector Addition Systems*. Our choice comes from that $Post^*$-saturation for *standard push* rules introduce fresh states (which lead infinite exploration), and the strict monotonicity validates Karp-Miller acceleration.

We write $\mathbb{N}_\omega$ for $\mathbb{N}\cup\{\omega\}$. Let us fix the dimension $k > 0$ and let $j(\boldsymbol{n})$ be the j-th element of a vector $\boldsymbol{n} \in \mathbb{N}_\omega^k$. The zero-vector is denoted by $\mathbf{0}$ with $j(\mathbf{0}) = 0$ for each $j \leq k$. A sequence of vectors is denoted with a tilde, like $\widetilde{\boldsymbol{n}}$. For $J \subseteq [1..k]$, we define the following orderings on vectors:

- $\boldsymbol{n} <_J \boldsymbol{n}'$ if $j(\boldsymbol{n}) < j(\boldsymbol{n}')$ for $j \in J$ and $j(\boldsymbol{n}) = j(\boldsymbol{n}')$ for $j \notin J$.
- $\boldsymbol{n} \leq_J \boldsymbol{n}'$ if $j(\boldsymbol{n}) \leq j(\boldsymbol{n}')$ for $j \in J$ and $j(\boldsymbol{n}) = j(\boldsymbol{n}')$ for $j \notin J$.
- $\boldsymbol{n}_1 \cdots \boldsymbol{n}_l \leqslant\!\!\leqslant_J \boldsymbol{n}'_1 \cdots \boldsymbol{n}'_{l'}$ if $l = l'$ and $\boldsymbol{n}_i \leq_J \boldsymbol{n}'_i$ for each $i \leq l$.
- $\boldsymbol{n}_1 \cdots \boldsymbol{n}_l \ll_J \boldsymbol{n}'_1 \cdots \boldsymbol{n}'_{l'}$ if $\boldsymbol{n}_1 \cdots \boldsymbol{n}_l \leqslant\!\!\leqslant_J \boldsymbol{n}'_1 \cdots \boldsymbol{n}'_{l'}$ and $\boldsymbol{n}_i <_J \boldsymbol{n}'_i$ for some i.

For example, $(1,2) <_{\{2\}} (1,3)$, $(1,2) \leq_{\{1,2\}} (1,3)$, $(1,2)(1,1) \leqslant\!\!\leqslant_{\{1,2\}} (1,3)(1,1)$, and $(1,2)(1,1) \not\ll_{\{1,2\}} (1,3)(1,1)$. We will omit J of $\leq_J$ if $J = \{1..k\}$.

If $\boldsymbol{n} <_J \boldsymbol{n}'$, an *acceleration* $\boldsymbol{n} \upharpoonright \boldsymbol{n}'$ is given by $\boldsymbol{n}_J^{\uparrow}$ where $j(\boldsymbol{n}_J^{\uparrow}) = \omega$ if $j \in J$, and $j(\boldsymbol{n}_J^{\uparrow}) = j(\boldsymbol{n})$ otherwise. For example, $(1,2) \upharpoonright (2,2) = (1,2)_{\{1\}}^{\uparrow} = (\omega, 2)$.

Definition 6. *Fix* $k \in \mathbb{N}$. *A* Pushdown Vector Addition Systems *(PDVAS) is a WSPDS* $\langle P, (\mathbb{N}^k, \leq), \Delta\rangle$ *where*

- P *is finite.*
- $\Delta \in P \times P \times \mathcal{P}Fun((\mathbb{N}^k)^{\leq 2}, \mathbb{N}^k)$ *is finite and without standard push rules.*
- ψ *is effectively computed and* strictly monotonic *wrt* $\ll_J$ *for each rule* $(p, q, \psi) \in \Delta$ *and* $J \subseteq [1..k]$.

Strict monotonicity wrt $\ll_J$ is crucial for acceleration, which naturally holds in VASs. A VAS transition $\boldsymbol{n} \hookrightarrow \boldsymbol{n}+\boldsymbol{z}$ holds $\boldsymbol{n}'+\boldsymbol{z} >_J \boldsymbol{n}+\boldsymbol{z}$ for each $\boldsymbol{n}' >_J \boldsymbol{n}$. A WSPDS may have a non-standard pop rule $(p, \boldsymbol{n}_1\boldsymbol{n}_2 \to q, \boldsymbol{m})$, and we require that the growth of either $\boldsymbol{n}_1$ or $\boldsymbol{n}_2$ leads the growth of $\boldsymbol{m}$.

4.1 Dependency

Acceleration for a VAS occurs when a descendant is strictly larger than some of its ancestors. However, for a PDVAS, such descendant-ancestor relation is not obvious in a P-automaton. We introduce *dependency* $\Rrightarrow$ on P-automata transitions $\mapsto$. The dependency is generated during $Post^*$-saturation steps.

Definition 7. *For a PDS $\langle P, \Gamma, \Delta\rangle$, a* dependency $\Rrightarrow$ *over transitions of a $Post^*$-automaton is generated during the saturation procedure, starting from $\emptyset$.*

1. *If a transition $p' \overset{\beta}{\mapsto} q$ is added from a rule $(p, \alpha \to p', \beta)$ and transition $p \overset{\alpha}{\mapsto} q$, then $(p \overset{\alpha}{\mapsto} q) \Rrightarrow (p' \overset{\beta}{\mapsto} q)$.*
2. *If a transition $p' \overset{\gamma}{\mapsto} q$ is added from a rule $(p, \alpha\beta \to p', \gamma)$ and transitions $p \overset{\alpha}{\mapsto} q' \overset{\beta}{\mapsto} q$, then $(p \overset{\alpha}{\mapsto} q') \Rrightarrow (p' \overset{\gamma}{\mapsto} q)$ and $(q' \overset{\beta}{\mapsto} q) \Rrightarrow (p' \overset{\gamma}{\mapsto} q)$.*
3. *Otherwise, we do not update $\Rrightarrow$.*

We denote the reflexive transitive closure of $\Rrightarrow$ by $\Rrightarrow^*$. Strict monotonicity leads to the following lemma, which guarantees the soundness of accelerations.

Lemma 3. *For a $Post^*$-automaton $\mathcal{A}$ of a PDVAS, if $p \overset{\boldsymbol{n}}{\mapsto} q \Rrightarrow^* p' \overset{\boldsymbol{m}}{\mapsto} q'$ and $p \overset{\boldsymbol{n}'}{\mapsto} q \in \nabla(\mathcal{A})$ for $\boldsymbol{n}' >_J \boldsymbol{n}$ hold, there exists $\boldsymbol{m}' >_J \boldsymbol{m}$ such that $p' \overset{\boldsymbol{m}'}{\Longmapsto} q' \in \nabla(\mathcal{A})$ and $p \overset{\boldsymbol{n}'}{\mapsto} q \Rrightarrow^* p' \overset{\boldsymbol{m}'}{\Longmapsto} q'$.*

Note that, if $(p \overset{\boldsymbol{n}}{\mapsto} q) \Rrightarrow^* (p \overset{\boldsymbol{n}_1}{\mapsto} q)$ and $\boldsymbol{n} <_J \boldsymbol{n}_1$ hold, Lemma 3 concludes

$$(p \overset{\boldsymbol{n}}{\mapsto} q) \Rrightarrow^* (p \overset{\boldsymbol{n}_1}{\mapsto} q) \Rrightarrow^* (p \overset{\boldsymbol{n}_2}{\mapsto} q) \Rrightarrow^* \cdots \Rrightarrow^* (p \overset{\boldsymbol{n}_i}{\mapsto} q) \Rrightarrow^* \cdots$$

with $\boldsymbol{n}_i <_J \boldsymbol{n}_{i+1}$ for each i. Thus, we can safely apply the acceleration on J.

4.2 $Post^*_F$-saturation

As in Section 4.1, accelerations will occur when $p \overset{\boldsymbol{n}}{\mapsto} q \Rrightarrow^* p \overset{\boldsymbol{n}'}{\mapsto} q$ and $\boldsymbol{n} <_J \boldsymbol{n}'$ is found for some p, q and J during the $Post^*$-saturation steps. We combine dependency generation and accelerations into the post saturation rules for a PDVAS. This new saturation procedure is denoted by $Post^*_F$, and a resulting P-automaton is called a $Post^*_F$-automaton.

We conservatively extend ψ in a PDVAS, from $(\mathbb{N}^k)^{\leq 2} \to \mathbb{N}^k$ to $(\mathbb{N}^k_\omega)^{\leq 2} \to \mathbb{N}^k_\omega$ by $\psi(\widetilde{\boldsymbol{n}}) = sup\{\psi(\widetilde{\boldsymbol{n}}') \mid \widetilde{\boldsymbol{n}}' \in (\mathbb{N}^k)^{\leq 2}, \widetilde{\boldsymbol{n}}' \leqslant\!\!\!\leqslant \widetilde{\boldsymbol{n}}\}$ for $\widetilde{\boldsymbol{n}} \in (\mathbb{N}^k_\omega)^{\leq 2}$,

Definition 8. *For a PDVAS $\langle P, (\mathbb{N}^k, \leq), \Delta \rangle$, let $\mathcal{A}_0 = (S_0, (\mathbb{N}^k_\omega, \leq), (\nabla_0, \emptyset), F)$ be an initial P-automaton accepting C_0. $Post^*_F(\mathcal{A}_0)$ is the result of repeated applications of the following $Post^*_F$ saturation rules.*

$$\frac{(S, \Gamma, (\nabla, \Rrightarrow), F), \quad p \overset{\widetilde{\boldsymbol{n}}}{\Longmapsto} q}{(S \cup \{p'\}, \Gamma, (\nabla, \Rrightarrow) \oplus (p' \overset{\boldsymbol{n}}{\mapsto} q, \Rrightarrow'), F)} \quad (p, p', \psi) \in \Delta,\ \psi(\widetilde{\boldsymbol{n}}) = \boldsymbol{n}$$

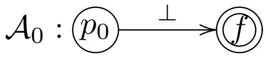
where $\Rrightarrow'$ is the dependency newly added by Definition 7.[3] *The operation $\oplus$ is defined as* $(\nabla, \Rrightarrow) \oplus (p' \overset{\boldsymbol{n}}{\mapsto} q, \Rrightarrow') =$

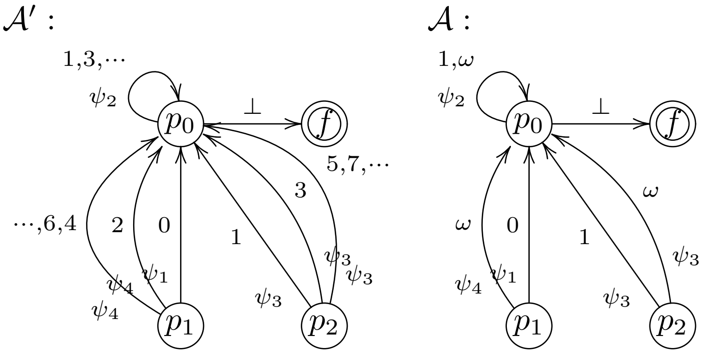
$$\begin{cases} (\nabla \cup \{p' \overset{\boldsymbol{n}' \upharpoonright \boldsymbol{n}}{\longmapsto} q\}, \Rrightarrow \cup \Rrightarrow'_{\upharpoonright}) & \textit{if there exists } p' \overset{\boldsymbol{n}'}{\mapsto} q \in \nabla \textit{ such that} \\ & p' \overset{\boldsymbol{n}'}{\mapsto} q \Rrightarrow^* \cdot \Rrightarrow' p' \overset{\boldsymbol{n}}{\mapsto} q \textit{ and } \boldsymbol{n}' <_J \boldsymbol{n} \textit{ for } J \neq \phi \\ (\nabla \cup \{p' \overset{\boldsymbol{n}}{\mapsto} q\}, \Rrightarrow \cup \Rrightarrow') & \text{otherwise} \end{cases}$$

where $\Rrightarrow'_{\upharpoonright}$ is obtained from $\Rrightarrow'$ by replacing its destination $p' \overset{\boldsymbol{n}}{\mapsto} q$ with $p' \overset{\boldsymbol{n}' \upharpoonright \boldsymbol{n}}{\longmapsto} q$.

Example 1. The following figure shows a $Post^*$-automaton $\mathcal{A}'$ and a $Post^*_F$-automaton $\mathcal{A}$ of a PDVAS with transition rules $\psi_1, \psi_2, \psi_3, \psi_4$. An initial configuration $C_0 = \{\langle p_0, \bot \rangle\}$ is accepted by $\mathcal{A}_0$. In $\mathcal{A}'$, $p_2 \overset{1}{\mapsto} p_0$ is generated from $p_1 \overset{0}{\mapsto} p_0 \overset{1}{\mapsto} p_0$ by ψ_3, and $p_1 \overset{2}{\mapsto} p_0$ is generated from $p_2 \overset{1}{\mapsto} p_0$ by ψ_4. Similarly, infinitely many $p_1 \overset{2k}{\Longmapsto} p_0$'s (and others) are generated. In $\mathcal{A}$, we have $(p_1 \overset{0}{\mapsto} p_0) \Rrightarrow (p_2 \overset{1}{\mapsto} p_0) \Rrightarrow (p_1 \overset{2}{\mapsto} p_0)$. An acceleration adds $(p_1 \overset{\omega}{\mapsto} p_0)$ instead of $(p_1 \overset{2}{\mapsto} p_0)$. Then, $p_2 \overset{\omega}{\mapsto} p_0$ and $p_0 \overset{\omega}{\mapsto} p_0$ are added by ψ_3 and ψ_2, respectively. This shows finitely convergence to $\mathcal{A}$, and we obtain $(post^*(C_0))^{\downarrow} = L(\mathcal{A})^{\downarrow} \cap (\mathbb{N}^k)^*$.

$\psi_1 : p_0, \epsilon \to p_1, 0$
$\psi_2 : p_1, \boldsymbol{n} \to p_0, \boldsymbol{n} + 1$
$\psi_3 : p_1, \boldsymbol{n}_1\boldsymbol{n}_2 \to p_2, \boldsymbol{n}_1 + \boldsymbol{n}_2$
$\psi_4 : p_2, \boldsymbol{n} \to p_1, \boldsymbol{n} + 1$





An immediate observation is that each configuration in $L(Post^*(\mathcal{A}_0))$ is covered by some in $L(Post^*_F(\mathcal{A}_0))$. The opposite follows from Lemma 4, which says that the downward closure (in $\mathbb{N}^k$) of a transition in $Post^*_F(\mathcal{A}_0)$ is included in the downward closure of transitions in $Post^*(\mathcal{A}_0)$. Its proof is found in [23].

Lemma 4. *For a PDVAS, let $\mathcal{A}_0$ be an initial P-automaton. If $p \overset{\boldsymbol{n}}{\mapsto} q$ is in $Post^*_F(\mathcal{A}_0)$, for each $\boldsymbol{n}' \leq \boldsymbol{n}$ with $\boldsymbol{n}' \in \mathbb{N}^k$, there exists $\boldsymbol{n}''$ such that $p \overset{\boldsymbol{n}''}{\Longmapsto} q$ is in $Post^*(\mathcal{A}_0)$ and $\boldsymbol{n}' \leq \boldsymbol{n}'' \leq \boldsymbol{n}$.*

[3] $\Rrightarrow' = \emptyset$ if (p, p', ψ) is a push rule; otherwise, the destination of $\Rrightarrow'$ is $p' \overset{\boldsymbol{n}}{\mapsto} q$.

Since a PDVAS does not have standard-push rules, the saturation procedure does not add new states. Thus, the sets of states in $Post^*_F(\mathcal{A}_0)$ and $Post^*(\mathcal{A}_0)$ are the same. From Lemma 4, we can obtain $L(Post^*_F(\mathcal{A}_0))^{\downarrow} \cap (\mathbb{N}^k)^* = (post^*(C_0))^{\downarrow}$.

Finite convergence of $Post^*_F$-saturation follows from that $\{(p, \boldsymbol{n}, q) \mid p, q \in S, \boldsymbol{n} \in \mathbb{N}^k_\omega\}$ is well-quasi-ordered. Thus, since accelerations can occur only finitely many times on a path of $\Rrightarrow^*$, the length of $\Rrightarrow^*$ is finite. Since $\Rrightarrow^*$ is finitely branching, König's lemma concludes that the $\Rrightarrow$-tree is finite.

Theorem 2. *For a PDVAS, if an initial P-automaton $\mathcal{A}_0$ with $L(\mathcal{A}_0) = C_0$ is finite, $Post^*_F(\mathcal{A}_0)$ finitely converges with $L(Post^*_F(\mathcal{A}_0))^{\downarrow} \cap (\mathbb{N}^k)^* = (post^*(C_0))^{\downarrow}$.*

4.3 Coverability of RVASS

In this section, we show that Recursive Vector Addition Systems with States (RVASSs) [3] are special cases of PDVASs, and Theorem refthm:termination implies decidability of its coverability.

Definition 9. *[3] Fix $k \in \mathbb{N}$. An* RVASS *$\langle Q, \delta\rangle$ consists of finite sets Q and δ of states and transitions, respectively. We denote*

- $q \xrightarrow{\boldsymbol{z}} q'$ *if* $(q, q', \boldsymbol{z}) \in \delta$ *for* $\boldsymbol{z} \in \mathbb{Z}^k$, *and*
- $q \xrightarrow{q_1 q_2} q'$ *if* $(q, q_1, q_2, q') \in \delta$.

The configuration $c \in (Q \times \mathbb{N}^k)^*$ represents a stack of pairs $\langle p, \boldsymbol{n}\rangle$ where $p \in Q$ and $\boldsymbol{n} \in \mathbb{N}^k$. The semantics is defined by following rules:

$$\frac{q \xrightarrow{\boldsymbol{z}} q' \quad \boldsymbol{n} + \boldsymbol{z} \in \mathbb{N}^k}{\langle q, \boldsymbol{n}\rangle c \hookrightarrow \langle q', \boldsymbol{n} + \boldsymbol{z}\rangle c} \qquad \frac{q \xrightarrow{q_1 q_2} q'}{\langle q, \boldsymbol{n}\rangle c \hookrightarrow \langle q_1, \boldsymbol{0}\rangle\langle q, \boldsymbol{n}\rangle c} \qquad \frac{q \xrightarrow{q_1 q_2} q'}{\langle q_2, \boldsymbol{n}'\rangle\langle q, \boldsymbol{n}\rangle c \hookrightarrow \langle q', \boldsymbol{n} + \boldsymbol{n}'\rangle c}$$

The *state-reachability* problem of an RVASS is, given two states q_0, q_f, whether there exist a vector $\boldsymbol{n}$ and a configuration c such that $\langle q_0, \boldsymbol{0}\rangle \hookrightarrow^* \langle q_f, \boldsymbol{n}\rangle c$. Lemma 3 in [3] showed its decidability by a reduction to a Branching VASS [6]. Below, Corollary 1 shows the decidability of the coverability. Note that the state reachability is the coverability from $\langle q_0, \boldsymbol{0}\rangle$ to $\{\langle q_f, \boldsymbol{0}^{\uparrow}\rangle\ \mathtt{any}^*\}$.

The encoding from an RVASS to a PDVAS is straightforward by regarding a configuration of an RVASS as a stack content in a PDVAS with a single control state $\bullet$, where $\langle q_i, (n_1, \cdots, n_k)\rangle \in Q \times \mathbb{N}^k$ is regarded as an element in $\Gamma = \mathbb{N}^{|Q|k}$

$$(\underbrace{0, \cdots, 0}_{(i-1)k}, n_1, \cdots, n_k, \underbrace{0, \cdots, 0}_{(|Q|-i)k})$$

Definition 10. *For $k \in \mathbb{N}$ and an* RVASS *$R = \langle Q, \delta\rangle$, a PDVAS $M_R = (\{\bullet\}, \Gamma, \Delta)$ consists of $\Gamma = \mathbb{N}^{|Q|k}$ and $\Delta \subseteq \{\bullet\} \times \{\bullet\} \times \mathcal{P}Fun(\Gamma^{\leq 2}, \Gamma)$ with*

1. *if $(q, q', \boldsymbol{z}) \in \delta$, then $(\bullet, \langle q, \boldsymbol{n}\rangle \to \bullet, \langle q', \boldsymbol{n} + \boldsymbol{z}\rangle) \in \Delta$.*
2. *if $(q, q_1, q_2, q') \in \delta$, then*
   (a) $(\bullet, \langle q, \epsilon\rangle \to \bullet, \langle q_1, \boldsymbol{0}\rangle) \in \Delta$ and (b) $(\bullet, \langle q_2, \boldsymbol{n}\rangle\langle q, \boldsymbol{m}\rangle \to \bullet, \langle q', \boldsymbol{n}+\boldsymbol{m}\rangle) \in \Delta$.

Corollary 1. *The coverability of an RVASS is decidable.*

5 *Pre**-automata for Coverability

When Δ has no non-standard pop rules, Pre^* does not introduce any fresh states, and we will show that ideal representations leads finite convergence. In this section, we assume that Δ has no non-standard pop rules.

5.1 Ideal Representation of *Pre**-automata

As mentioned in Section 3.2, we need to construct a Pre^*-automaton that accepts predecessors of an ideal $C_0^{\uparrow}$. A naive representation of such upward closures may be infinite. Therefore, we use an ideal representation Pre_F^*-automaton in which transition labels and states are ideals. Thanks to WQO, an ideal is characterized by its finitely many minimal elements, and ideals are well founded wrt set inclusion.

Definition 11. *For a WSPDS* $\langle (P, \preceq), (\Gamma, \leq), \Delta \rangle$*, by replacing* Γ *with* $\mathcal{I}(\Gamma)$ *and* $P \subseteq S \setminus F$ *with* $\mathcal{I}(P) \subseteq S \setminus F$ *in Definition 3, we obtain the definition of a* Pre_F^**-automaton* $\mathcal{A} = (S, \mathcal{I}(\Gamma), \nabla, F)$.

As notational convention, let s, t to range over S, ideals K, K' to range over $\mathcal{I}(P)$, and I, I' over $\mathcal{I}(\Gamma)$. We denote $w \in \widetilde{I}$ for $\widetilde{I} = I_1 I_2 \cdots I_n$, if $w = \alpha_1 \alpha_2 \cdots \alpha_n$ and $\alpha_i \in I_i$ for each i. We say that $\mathcal{A}$ accepts a configuration $\langle p, w \rangle$, if there is a path $K \overset{\widetilde{I}}{\Longmapsto} f \in F$ in $\mathcal{A}$ and $p \in K$, $w \in \widetilde{I}$. The ideal representation of an initial P-automaton accepting $C_0^{\uparrow}$ is obtained from a P-automaton accepting C_0 by replacing each state p with $\{p\}^{\uparrow}$ and each transition label α with $\{\alpha\}^{\uparrow}$.

Definition 12. *Let* $\mathcal{A}_0$ *be an initial* Pre_F^**-automaton accepting* $C_0^{\uparrow}$*.* $Pre_F^*(\mathcal{A}_0)$ *is the result of repeated applications of the following* Pre_F^**-saturation rules*

$$\frac{(S, \mathcal{I}(\Gamma), \nabla, F), \quad K \overset{\widetilde{I}}{\Longmapsto} s}{(S, \mathcal{I}(\Gamma), \nabla, F) \oplus \{\phi^{-1}(K) \overset{\psi^{-1}(\widetilde{I})}{\longmapsto} s\}} \ \textit{if } \widetilde{I} \in \mathcal{I}(\Gamma^{\leq 2}) \textit{ and } (\phi, \psi) \in \Delta$$

where $\phi^{-1}(K) \neq \emptyset$*,* $\psi^{-1}(\widetilde{I}) \neq \emptyset$*, and* $(S, \Sigma, \nabla, F) \oplus \{K \overset{I}{\mapsto} s\}$ *is*

$$\begin{cases} (S, \Sigma, \nabla, F) & \textit{if } (K' \overset{I'}{\mapsto} s) \in \nabla \textit{ with } K \subseteq K' \textit{ and } I \subseteq I' \\ (S, \Sigma, (\nabla \setminus \{K \overset{I'}{\mapsto} s\}) \cup \{K \overset{I' \cup I}{\longmapsto} s\}, F) & \textit{if } (K \overset{I'}{\mapsto} s) \in \nabla \\ (S \cup \{K\}, \Sigma, \nabla \cup \{K \overset{I}{\mapsto} s\}, F) & \textit{otherwise} \end{cases}$$

The $\oplus$ operator merges ideals associated to transitions. Assume that a new transition $K \overset{I}{\mapsto} s$ is generated. If there is a transition $K' \overset{I'}{\mapsto} s$ with the same s, $K \subseteq K'$, and $I \subseteq I'$, the ideal of configurations starting from $K \overset{I}{\mapsto} s$ is included in that from $K' \overset{I'}{\mapsto} s$. Thus, no needs to add it. If there is a transition $K \overset{I'}{\mapsto} s$ between the same pair K, s, then take the union $I \cup I'$. Otherwise, we add a new transition.

It is easy to see that if $\phi \in \mathcal{P}Fun(X,Y)$ is monotonic, then, for any $I \in \mathcal{I}(Y)$, $\phi^{-1}(I)$ is an ideal in $\mathcal{I}(X)$. Completeness $pre^*(C_0^{\uparrow}) \subseteq L(Pre_F^*(\mathcal{A}_0))$ follows immediately by induction on saturation steps. Soundness $pre^*(C_0^{\uparrow}) \supseteq L(pre^*(\mathcal{A}_0))$ is guaranteed by Lemma 5, which is an invariant during the saturation procedure.

Lemma 5. *Assume* $K \overset{\widetilde{I}}{\longmapsto} s$ *in* $Pre_F^*(\mathcal{A}_0)$*. For each* $p \in K$*,* $w \in \widetilde{I}$*,*

- *if* $s = K' \in \mathcal{I}(P)$*, then* $\langle p, w\rangle \hookrightarrow^* \langle q, \epsilon\rangle$ *for some* $q \in K'$*.*
- *if* $s \notin \mathcal{I}(P)$*, there exists* $K' \overset{\widetilde{I'}}{\longmapsto} s$ *in* $\mathcal{A}_0$ *such that* $\langle p, w\rangle \hookrightarrow^* \langle p', w'\rangle$ *for some* $p' \in K'$ *and* $w' \in \widetilde{I'}$*.*

Theorem 3. *For an initial P-automaton* $\mathcal{A}_0$ *accepting* $C_0^{\uparrow}$*,* $L(Pre_F^*(\mathcal{A}_0)) = pre^*(C_0^{\uparrow})$*.*

Note that Theorem 3 only shows the correctness of Pre_F^*-saturation. We do not assume its finite convergence, which will be discussed in next two subsections.

5.2 Coverability of Multi-set PDS

As an example of the finite convergence, we show *Multi-set pushdown system* (Multi-set PDS) proposed in [20,13], which is an extension of PDS by attaching a multi-set into the configuration. We directly give the definition of a Multi-set PDS as a WSPDS. Note that, although a Multi-set PDS has infinitely many control states, it finitely converges because of restrictions on decreasing rules.

Definition 13. *A* Multi-set pushdown system (Multi-set PDS) *is a WSPDS* $((Q \times \mathbb{N}^k, \preceq), \Gamma, \delta)$*, where*

- Q*,* Γ *are finite and* $k = |\Gamma|$*,*
- δ *is a finite set of transition rules consisting of two kinds:*
  1. *Increasing rules* $\delta_1 : (p, \gamma, q, w, \boldsymbol{n})$ *for* $\boldsymbol{n} \in \mathbb{N}^k$*;*
  2. *Decreasing rules* δ_2*:* $(p, \perp, q, \perp, \boldsymbol{n})$ *for* $\boldsymbol{n} \in \mathbb{N}^k$*.*

Configuration transitions are defined by:

$$\frac{(p,\gamma,q,w,\boldsymbol{n}) \in \delta_1}{\langle (p,\boldsymbol{m}), \gamma w'\rangle \hookrightarrow \langle (q, \boldsymbol{n}+\boldsymbol{m}), ww'\rangle} \qquad \frac{(p,\perp,q,\perp,\boldsymbol{n}) \in \delta_2, \boldsymbol{m} \geq \boldsymbol{n}}{\langle (p,\boldsymbol{m}), \perp\rangle \hookrightarrow \langle (q, \boldsymbol{m}-\boldsymbol{n}), \perp\rangle}$$

Note the decreasing rules are applied only when the stack is empty. A state in Pre_F^*-automata is in $\mathcal{I}(Q \times \mathbb{N}^k)$. Since Q is finite, we can always separate one state into finitely many states such that each of which has the form of $Q \times \mathcal{I}(\mathbb{N}^k)$. From Definition 12, we have two observations.

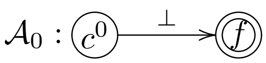
1. If transition $(p, K) \overset{\gamma}{\mapsto} s$ is added from $(q, K') \overset{w}{\longmapsto} s$ by an increasing rule in δ_1, then $K \supseteq K'$.
2. If transition $(p, K) \overset{\perp}{\mapsto} s$ is added from $(q, K') \overset{\perp}{\mapsto} s$ by a decreasing rule in δ_2, then $K \subseteq K'$ and s is a final state.

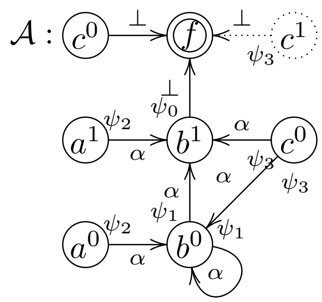
Pre^*_F-saturation steps by increasing rules always enlarge ideals of vectors. By Lemma 1, eventually such ideals become maximal. Since stack alphabet is (finite thus) well-quasi-ordered, newly generated transitions by increasing rules are eventually caught by the first case of the $\oplus$ operator (in Definition 12). A worrying case is by decreasing rules, which shrink ideals. Since WQO does not guarantee the stabilization for $I_0 \supset I_1 \supset \cdots$, it may continue infinitely. For instance, Pre^*_F-saturation steps by decreasing pop rules may expand a path $\mapsto^*$ endlessly. Fortunately, decreasing rules of a Multi-set PDS occur only when the stack is empty. In such cases, destination states of $\mapsto$ are always final states, which are finitely many. Therefore, again they are eventually caught by the first case of the $\oplus$ operator. Note that this argument works even if we relax finite stack alphabet in Definition 13 to being well-quasi-ordered.

Corollary 2. *The* coverability problem *for a Multi-set PDS (with well-quasi-ordered stack alphabet) is decidable.*

Example 2. Let $\langle (\{a,b,c\} \times \mathbb{N}, \preceq), \{\alpha\}, \delta\rangle$ be a Multi-set PDS with transition rules given below. The set of configurations covering $\langle c^0, \bot\rangle$ is computed by Pre^*_F-automaton $\mathcal{A}$. We abbreviate ideal $\{p^n\}^\uparrow$ by p^n for $p \in \{a,b,c\}$ and $n \geq 0$. A transition $c^1 \overset{\bot}{\mapsto} f$ is generated from $a^1 \overset{\alpha\bot}{\Longmapsto} f$ by ψ_3. However, it is not added since we already have $c^0 \overset{\bot}{\mapsto} f$ and $\{c^1\}^\uparrow \subseteq \{c^0\}^\uparrow$.

$$\begin{aligned}\delta_1 = \{\ &\psi_1 : (b^{\boldsymbol{n}}, \alpha \to a^{\boldsymbol{n}+1}, \alpha),\\ &\psi_2 : (a^{\boldsymbol{n}}, \alpha \to b^{\boldsymbol{n}}, \epsilon),\\ &\psi_3 : (c^{\boldsymbol{n}}, \epsilon \to a^{\boldsymbol{n}}, \alpha)\}\\ \delta_2 = \{\ &\psi_0 : (b^{\boldsymbol{n}}, \bot \to c^{\boldsymbol{n}-1}, \bot)\}\end{aligned}$$





5.3 Finite Control States

Assume that, for a monotonic WSPDS $M = \langle P, (\Gamma, \leq), \Delta\rangle$, P is finite and Δ does not contain nonstandard-pop rules. Then, we observe that, in the Pre^*_F-saturation for M, i) the set of states is bounded by the state in $\mathcal{A}_0$ and P, and ii) transitions between any pair of states are finitely many by Lemma 1. Hence, Pre^*_F saturation procedure finitely converges.

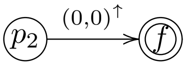
Theorem 4. *Let $\langle P, (\Gamma, \leq), \Delta\rangle$ be a WSPDS such that P is finite and $\psi^{-1}(I)$ is computable for any $(p, p', \psi) \in \Delta$. Then, its coverability is decidable.*

Example 3. Let $M = \langle \{p_i\}, \mathbb{N}^2, \Delta\rangle$ be a WSPDS with $\Delta = \{\psi_1, \psi_2, \psi_3, \psi_4\}$ given in the figure. An automaton $\mathcal{A}$ illustrates the *pre**-saturation starting from initial $\mathcal{A}_0$ that accepts $C = \langle p_2, (0,0)^\uparrow\rangle$.

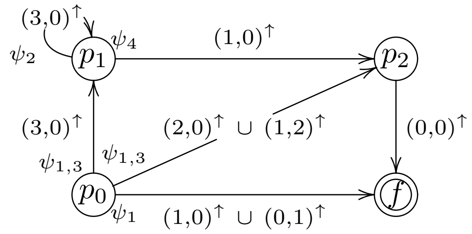
For instance, $p_1 \overset{(3,0)^\uparrow}{\longmapsto} p_1$ in $\mathcal{A}$ is generated by ψ_2, and $p_0 \overset{(3,2)^\uparrow}{\longmapsto} p_1$ is added by ψ_3. Then repeatedly apply ψ_1 twice to $p_0 \overset{(3,2)^\uparrow}{\longmapsto} p_1 \overset{(3,0)^\uparrow}{\longmapsto} p_1$, we obtain $p_0 \overset{(3,0)^\uparrow}{\mapsto} p_1$.

$\psi_1 : \langle p_0, \boldsymbol{n} \rangle \to \langle p_0, (\boldsymbol{n} + (1,1))\boldsymbol{n} \rangle$
$\psi_2 : \langle p_1, \boldsymbol{n} \rangle \to \langle p_1, \epsilon \rangle$ if $\boldsymbol{n} \geq (3,0)$
$\psi_3 : \langle p_0, \boldsymbol{n} \rangle \to \langle p_1, \boldsymbol{n} - (0,2) \rangle$ if $\boldsymbol{n} \geq (0,2)$
$\psi_4 : \langle p_1, \boldsymbol{n} \rangle \to \langle p_2, \epsilon \rangle$ if $\boldsymbol{n} \geq (1,0)$

$\mathcal{A}_0$:



$\mathcal{A}$:



6 Conclusion

This paper investigated *well-structured pushdown systems* (WSPDSs), pushdown systems with well-quasi-ordered control states and stack alphabet, and developed two proof techniques to investigate the coverability based on extensions of classical P-automata techniques. They are,

- when a WSPDS has no standard push rules, the forward P-automata construction $Post^*$ with Karp-Miller acceleration, and
- when a WSPDS has no non-standard pop rules, the backward P-automata construction Pre^* with ideal representations.

We showed decidability results of coverability under certain conditions, which include *recursive vector addition system with states* [3], *multi-set pushdown systems* [20,13], and a WSPDS with finite control states and WQO stack alphabet. The first one extended the decidability of the state reachability in [3] to that of the coverability, and the second one relaxed finite stack alphabet of Multi-set PDSs [20,13] to being well-quasi-ordered.

Our current results just opened the possibility of WSPDSs. Among lots of things to do, we list few for future works.

- Currently, we have few examples of WSPDSs. For instance, parameterized systems would be good candidates to explore.
- Currently, we are mostly investigating with finite control states. However, we also found that a naive extension to infinite control states weakens the results a lot. We are looking for alternative conditions.
- Our decidability proofs contain algorithms to compute, however the estimation of their complexity is not easy due to the nature of WQO. We hope that a general theoretical observation [22] would give some hints.
- Our current forward method is restricted to VASs. We also hope to apply Finkel and Goubault-Larrecq's work on ω^2-WSTS [11,12] to generalize.

Acknowledgements. The authors would like to thank Prof.Alain Finkel and anonymous referees for valuable comments. This work is supported by the NSFC-JSPS bilateral joint research project (61011140074), NSFC projects (61003013,61100052,61033002), NSFC-ANR joint project (61261130589), and JSPS KAKENHI Grant-in-Aid for Scientific Research(B) (23300008).

References

1. Abdulla, P., Cerans, K., Jonsson, C., Yih-Kuen, T.: Algorithmic analysis of programs with well quasi-ordered domains. Information and Computation 160(1-2), 109–127 (2000)
2. Atig, M.F., Bouajjani, A., Qadeer, S.: Context-bounded analysis for concurrent pro-grams with dynamic creation of threads. In: Kowalewski, S., Philippou, A. (eds.) TACAS 2009. LNCS, vol. 5505, pp. 107–123. Springer, Heidelberg (2009)
3. Bouajjani, A., Emmi, M.: Analysis of recursively parallel programs. In: Principles of Programming Languages (POPL 2012), pp. 203–214. ACM (2012)
4. Bouajjani, A., Esparza, J., Maler, O.: Reachability analysis of pushdown automata: Application to model-checking. In: Mazurkiewicz, A., Winkowski, J. (eds.) CONCUR 1997. LNCS, vol. 1243, pp. 135–150. Springer, Heidelberg (1997)
5. Chadha, R., Viswanathan, M.: Decidability results for well-structured transition systems with auxiliary storage. In: Caires, L., Vasconcelos, V.T. (eds.) CONCUR 2007. LNCS, vol. 4703, pp. 136–150. Springer, Heidelberg (2007)
6. Demri, S., Jurdziński, M., Lachish, O., Lazic, R.: The covering and boundedness problems for branching vector addition systems. Journal of Computer and System Sciences 79(1), 23–38 (2012)
7. Esparza, J., Hansel, D., Rossmanith, P., Schwoon, S.: Efficient algorithms for model checking pushdown systems. In: Emerson, E.A., Sistla, A.P. (eds.) CAV 2000. LNCS, vol. 1855, pp. 232–247. Springer, Heidelberg (2000)
8. Finkel, A.: A generalization of the procedure of Karp and Miller to well structured transition systems. In: Ottmann, T. (ed.) ICALP 1987. LNCS, vol. 267, pp. 499–508. Springer, Heidelberg (1987)
9. Finkel, A., Willems, B., Wolper, P.: A direct symbolic approach to model checking pushdown systems. Electronic Notes Theoretical Computer Science 9, 27–37 (1997)
10. Finkel, A., Schnoebelen, P.: Well-structured transition systems everywhere! Theoretical Computer Science 256(1-2), 63–92 (2001)
11. Finkel, A., Goubault-Larrecq, J.: Forward analysis for WSTS, Part I: Completions. In: STACS 2009, pp. 433–444 (2009), `http://www.stacs-conf.org`
12. Finkel, A., Goubault-Larrecq, J.: Forward analysis for WSTS, Part II: Complete WSTS. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009, Part II. LNCS, vol. 5556, pp. 188–199. Springer, Heidelberg (2009)
13. Jhala, R., Majumdar, R.: Interprocedural analysis of asynchronous programs. In: Principles of Programming Languages (POPL 2007), pp. 339–350. ACM (2007)
14. Lazić, R.: The reachability problem for vector addition systems with a stack is not elementary (2012), `http://rp12.labri.fr` (manuscript)
15. Leroux, J.: Vector addition system reachability problem. In: Principles of Programming Languages (POPL 2011), pp. 307–316. ACM (2011)
16. Mayr, E.: An algorithm for the general Petri net reachability problem. SIAM Journal Computing 13(3), 441–460 (1984)
17. Mayr, R.: Process rewrite systems. Information and Computation 156, 264–286 (1999)
18. Qadeer, S., Rehof, J.: Context-bounded model checking of concurrent software. In: Halbwachs, N., Zuck, L.D. (eds.) TACAS 2005. LNCS, vol. 3440, pp. 93–107. Springer, Heidelberg (2005)

19. Ramalingam, G.: Context-sensitive synchronization-sensitive analysis is undecidable. ACM Trans. Programming Languages and Systems 22(2), 416–430 (2000)
20. Sen, K., Viswanathan, M.: Model checking multithreaded programs with asynchronous atomic methods. In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 300–314. Springer, Heidelberg (2006)
21. Verma, K., Goubault-Larrecq, J.: Karp-Miller trees for a branching extension of VASS. Discrete Mathematics & Theoretical Computer Science 7(1), 217–230 (2005)
22. Weiermann, A.: Complexity bounds for some finite form of Kruskal's theorem. Journal of Symbolic Computation 18, 463–488 (1994)
23. Xiaojuan, C., Ogawa, M.: Well-structured pushdown system, Part 1: Decidable classes for coverability. JAIST Research Report IS-RR-2013-001 (2013), `http://hdl.handle.net/10119/11347`