The Termination/Boundedness Problem of Well-Structured Pushdown Systems

Suhua Lei\textsuperscript{1}, Xiaojuan Cai\textsuperscript{1}, and Mizuhito Ogawa\textsuperscript{2}

\textsuperscript{1} BASICS Lab, Shanghai Jiao Tong University, China
leisuhua, cxj@sjtu.edu.cn
\textsuperscript{2} Japan Advanced Institute of Science and Technology, Japan
mizuhito@jaist.ac.jp

Abstract. Models that combine vector addition systems and pushdown systems have attracted lots of attentions recently. They can model some concurrent recursive computations, and at the same time enjoy some decidability properties. Among these models, well-structured pushdown system (WSPDS) \cite{1} is the most general one. It extends pushdown systems with the sets of states and stack alphabet being well-quasi-orders, and the decidability of most of its problems still remains unknown. Inspired by the work \cite{2} of Leroux et. al., we prove that the termination and boundedness problems of WSPDS are decidable, and the lower and upper bounds are both Hyper-Ackermannian for a subclass of WSPDS. Comparing with \cite{2}, we generalize from a finite stack alphabet to a (possibly infinite) well-quasi-ordered one, and our results show that the infiniteness of stack alphabet does not affect the decidability and even the complexity of the termination and boundedness problems.

1 Introduction

A pushdown system is a finite state system equipped with a stack for storing words over stack alphabet. It has recently been combined with vector addition systems by Sen et. al \cite{3} and Bouajani et. al \cite{4} to model multi-thread recursive programs. WSPDS is more general, extending pushdown system with the sets of states and stack alphabet being well-quasi-orders. Models in \cite{3} and \cite{4} are subclasses of WSPDS. We expect WSPDS to be more expressive and simultaneously enjoy some decidable properties. In this paper,

- we prove that the termination problem of WSPDS and the boundedness problem of strict WSPDS are decidable by extending the standard reachability tree technique,

- we show that the lower and upper bounds of these problems are both Hyper-Ackermannian for a subclass of WSPDS.

WSPDS is proposed in \cite{1}, and the technique used in this paper extends from \cite{2}. Comparing with these two work, in this paper,

- we generalize the model in \cite{2} from finite stack alphabet to well-quasi-ordered stack alphabet.
- instead of seeking subclasses of WSPDS with decidable coverability [1], which is a more difficult problem than termination/boundedness, we prove the decidability of the termination/boundedness problem of WSPDS.

\section{Well-structured pushdown systems}

A quasi-order \((X, \leq)\) is a reflexive and transitive relation \(\leq\) on \(X\). It is a well-quasi-order if any infinite sequence over \(X\) contains an increasing pair. A partial order is an antisymmetric quasi-order. In a partial order \((X, \leq)\), we write \(x < y\) if \(x \leq y\) and \(x \neq y\).

Since the transition rules manipulate words over stack alphabet \((\Gamma, \leq)\), we sometimes write \((p, w \rightarrow p', w')\) if there exists \(\psi \in \Delta\) and \(\psi(p, w) = (p', w')\).

\begin{definition}
A WSPDS is a triplet \(M = \langle (P, \leq), (\Gamma, \leq), \Delta \rangle\) where
\begin{itemize}
\item \((P, \leq)\) and \((\Gamma, \leq)\) are well-quasi-orders, and
\item \(\Delta \subseteq \mathcal{F}(P \times \Gamma^{\leq 2}, P \times \Gamma^{\leq 2})\) is the finite set of monotonic transition rules (w.r.t. \(\leq\)), where \(\mathcal{F}(X, Y)\) denotes the set of partial functions from \(X\) to \(Y\). We use \(\langle x \rangle\) to denote the top element, and \(|w|\) the length of \(w\). We define the head of a configuration as \(h(p, w) = (p, w[1])\) if \(w \neq \epsilon\), otherwise, \(h(p, w) = (p, \bot)\). If we require \(\forall \alpha \in I, \bot \leq \alpha\), the set of heads is a well-quasi-order over \(\leq= \leq \times \leq\).
\end{itemize}

A WSPDS is strict if \((P, \leq)\) and \((\Gamma, \leq)\) are partial orders, and the transition functions are strictly monotonic.

A configuration \(c\) is a pair \((p, w)\), where \(p\) is a state and \(w\) is a stack word. We use \(w[1]\) to denote the top element, and \(|w|\) the length of \(w\). We define the head of a configuration as \(h(p, w) = (p, w[1])\) if \(w \neq \epsilon\), otherwise, \(h(p, w) = (p, \bot)\). If we require \(\forall \alpha \in I, \bot \leq \alpha\), the set of heads is a well-quasi-order over \(\leq= \leq \times \leq\).

A run starting from the initial configuration \(c_0\) is a (finite or infinite) sequence \(c_0, c_1, \ldots\) of configurations, where \(c_{i-1} \rightarrow c_i\) for every index \(i > 0\). The reachability set of a run is the set of all configurations that occur on this run.

- **Termination** asks whether all runs starting from \(c_0\) are finite.
- **Boundedness** asks whether the reachability set of every run from \(c_0\) is finite.

\begin{example}
\(M = \langle (\mathbb{N}, \leq), (\mathbb{N}, \leq), \Delta \rangle\) is a WSPDS with both control states and stack alphabet being natural numbers. It is also a strict WSPDS since every transition function is strictly monotonic.

\[
\Delta = \begin{cases}
    r_1: p, \alpha \rightarrow p + 1, (\alpha - 1)(\alpha - 1) \\
r_2: p, \epsilon \rightarrow p + 2, 0 \quad \text{if} \ p \geq 2 \\
r_3: p, \alpha \rightarrow p - 3, \alpha + 3 \\
r_4: p, \alpha \beta \rightarrow p, \alpha + \beta - 2
\end{cases}
\]

Assume \(c_0 = \langle 1, 1 \rangle\), this is an infinite run with infinite reachability set:

\[
\langle 1, 1 \rangle \xrightarrow{r_1} \langle 2, 0 \rangle \xrightarrow{r_2} \langle 2, 0, 0 \rangle \xrightarrow{r_3} \langle 4, 0, 0 \rangle \xrightarrow{r_4} \langle 1, 300 \rangle \xrightarrow{r_3} \langle 1, 10 \rangle \xrightarrow{r_1} \langle 2, 0 \rangle \cdots
\]

If we change rule \(r_1\) to \(p, \alpha \rightarrow p + 1, (\alpha - 1)\), we can get an infinite run with finite reachability set: \(\langle 1, 1 \rangle \xrightarrow{r_1} \langle 2, 0 \rangle \xrightarrow{r_2} \langle 2, 0, 0 \rangle \xrightarrow{r_3} \langle 4, 0, 0 \rangle \xrightarrow{r_4} \langle 1, 300 \rangle \xrightarrow{r_3} \langle 1, 1 \rangle \xrightarrow{r_1} \langle 2, 0 \rangle \cdots\).

And without rule \(r_2\), all runs starting from \(c_0\) terminate.
3 Termination/Boundedness problem

The reduced reachability tree is a standard technique for vector addition systems. Leroux et al. extend it to solve the termination/boundedness problem for vector addition systems equipped with a stack. We extend it further to WSPDS. The reachability tree of a WSPDS with initial configuration $c_0$ is a directed unordered tree with root $r : c_0$, and each node $n : c_n$ has child $m : c_m$ if $c_n \hookrightarrow c_m$.

Termination problem

In this part, we give a necessary and sufficient condition for non-termination. This condition can be checked in a finite prefix of the reachability tree, which implies the decidability of termination problem.

Definition 2. A node $s : \langle p, w \rangle$ pumps a node $t : \langle q, v \rangle$ if
- there is a path from $s$ to $t$, and every node $t' : \langle p', w' \rangle$ on it satisfies $|w'| \geq |w|$.
- $h(\langle p, w \rangle) \sqsubseteq h(\langle q, v \rangle)$, i.e., $p \preceq q$ and either $w = \epsilon$ or $w[1] \leq v[1]$.

We call a node pumpable if there exists some node that pumps it. The notion of pumpable nodes is similar to subsumed nodes in [2], but we consider the increasing of heads, other than just states. Intuitively, the first condition of Definition 2 means that the run from $\langle p, w \rangle$ to $\langle q, v \rangle$ never touches the contents below $w[1]$, and the second implies that configuration $\langle q, v \rangle$ with larger head than $\langle p, w \rangle$ can simulate this run and pump a configuration with even larger head.

Conversely, assume $\langle p_0, w_0 \rangle \hookrightarrow \langle p_1, w_1 \rangle \cdots$ is an infinite run, we can extract an infinite subsequence of configurations from it, say $\langle p_{i_0}, w_{i_0} \rangle, \langle p_{i_1}, w_{i_1} \rangle, \cdots$, each time choosing the configuration with the minimal stack length among the configurations behind the last chosen configuration. Note that each pair $\langle p_{i_k}, w_{i_k} \rangle$ and $\langle p_{i_j}, w_{i_j} \rangle$ with $k < j$ in this subsequence satisfies the first condition of pumpable nodes. By the fact that the set of heads is a well-quasi-order over $\succeq$, the infinite subsequence we obtain above must contain a pumpable node.

Let the reduced reachability tree be the largest prefix of the reachability tree such that every pumpable node has no child. The following theorem provides an algorithm to decide the termination of WSPDS.

Theorem 1. A WSPDS has an infinite run if, and only if, its reduced reachability tree contains a pumpable node.

Boundedness problem

If a pumpable node is exactly the same as the one that pumps it, then the infinite run induced by this pumpable node will have a finite reachability set. Therefore, we need more than a pumpable node for the boundedness problem.

Definition 3. A node $s : \langle p, w \rangle$ strictly pumps a node $t : \langle q, v \rangle$ if $s$ pumps $t$ and either $|w| < |v|$ or $h(\langle p, w \rangle) \sqsubset h(\langle q, v \rangle)$.
In Example 1, configuration \( (1, 1) \) strictly pumps \( (1, 10) \). However, with modified rule \( r1 \), \( (1, 1) \) does not strictly pump \( (1, 1) \).

**Theorem 2.** A strict WSPDS has an infinite reachability set if, and only if, its reduced reachability tree contains a strictly pumpable node.

The proof is similar to theorem 1. We need the strictness of WSPDS during the proof because only in a partial order can we apply a strict order on heads of configurations and only strictly monotonic transition rules can guarantee the strict growth of configurations.

4 Complexity: lower and upper bounds

The estimation on the size of reduced reachability tree is difficult for any well-quasi-orders. We make the following assumption for WSPDS in this part:

- the well-quasi-orders are restricted to vectors, i.e., \( P = \mathbb{N}^d \) and \( \Gamma = \mathbb{N}^k \).
- the changes of vectors caused by one-step computation should be controlled: each non-pop rule \( (p, w, q, v) \in \Delta \) satisfies \( \|q - p\|_\infty \leq 1 \) and \( \|\Sigma(v) - \Sigma(w)\|_\infty \leq 1 \), where \( \|p\|_\infty \) maps a vector \( p \) to its maximum component, and \( \Sigma(w) = \sum_{i \in \{1..n\}} w[i] \) if \( w = w[1] \cdots w[n] \).

With these restrictions, any run can be mapped to a \( n \)-controlled nested sequence, and the height of a reduced reachability tree is the maximal length of \( n \)-controlled bad nested sequence, which is proved to be Hyper-Ackermannian [2]. Given a WSPDS \( A = (\mathbb{N}^d, \leq, \mathbb{N}^k, \leq, \Delta) \) with initial configuration \( (p_0, w_0) \), the size of \( A \) can be defined as

\[
|A| = d + k + (d + k) \cdot \max\{\|p_0\|_\infty, \|\Sigma(w_0)\|_\infty\} + (d + k) \cdot |\Delta|,
\]

**Theorem 3.** The reduced reachability tree of this subclass of WSPDS has at most \( F_\omega(|A|) \) nodes, where \( F_\omega \) is Hyper-Ackermannian function defined by diagonalization over the ordinals \( \alpha < \omega^\omega \).

The lower bound of the size of reduced reachability tree follows the same paper [2]. Leroux et. al. have proved the lower bound of the reduced reachability tree is Hyper-Ackermannian for their model, which is a subclass of the restricted WSPDS in this section.

**References**