Pushdown Systems with Stack Manipulation

Yuya Uezato and Yasuhiko Minamide

University of Tsukuba
uezato@score.cs.tsukuba.ac.jp,
minamide@cs.tsukuba.ac.jp

Abstract. Pushdown systems are a model of computation equipped with one stack where only the top of the stack is inspected and modified in each step of transitions. Although this is a natural restriction, some extensions of pushdown systems require more general operations on stack: conditional pushdown systems inspect the whole stack contents and discrete timed pushdown systems increment the ages of the whole stack contents.

In this paper, we present a general framework called pushdown systems with transductions (TrPDS) for extending pushdown systems with transitions that modify the whole stack contents with a transducer. Although TrPDS is Turing complete, it is shown that if the closure of transductions appearing in the transitions of a TrPDS is finite, it can be simulated by an ordinary pushdown system and thus the reachability problem is decidable. Both of conditional and discrete timed pushdown systems can be considered as such subclasses of TrPDS.

1 Introduction

The theory of pushdown systems (PDS) has been successfully applied to the verification of recursive programs such as Java programs [4]. The essential result is that the reachability problem of a PDS can be decided efficiently by representing the rational (regular) set of configurations with automata [5,10,8]. Several extensions of PDS have been studied to widen the applications of PDS, and their reachability problems can often be decided by translating them to ordinary pushdown systems.

Esparza et al. introduced pushdown systems with checkpoints that can check the whole stack contents against a rational language [9] to model runtime stack inspection used for security checks. They showed that pushdown systems with checkpoints can be translated into ordinary pushdown systems, and thus the reachability problem is decidable. We call this extension *conditional pushdown systems* in this paper [12].

Abdulla et al. introduced *discrete timed pushdown systems* [2] that combine timed automata [3] and pushdown systems. Stack symbols of a timed pushdown system are extended with the notion of ages, and timed pushdown systems have a transition that increments the ages of all the symbols in the stack. Even with this extension, timed pushdown systems can also be translated into pushdown systems to decide the reachability problem.

D. Van Hung and M. Ogawa (Eds.): ATVA 2013, LNCS 8172, pp. 412–426, 2013.

In this paper, we generalize these extensions and present a general framework called *pushdown systems with transductions* (TrPDS) for extending pushdown systems with transitions that modify the whole stack contents with a finite-state transducer. Transductions are the relations induced by transducers. Since TrPDS is Turing complete in general, we are interested in a *finite* TrPDS where we impose the restriction that the closure of transductions appearing in the transitions of a TrPDS is finite. Both of conditional and timed pushdown systems can be formulated as simple instances of finite TrPDS. We then show that a *finite* TrPDS can be translated into an ordinary PDS by generalizing the construction of Abdulla et al. for timed pushdown systems [2] and the reachability problem of a finite TrPDS is decidable. As a nontrivial example of finite TrPDS, we introduce conditional transformable pushdown systems that can check the whole stack contents against a rational language and modify the whole stack contents by a function from stack symbols to stack symbols.

We also show that the saturation procedure that calculates $\mathrm{pre}^*(C)$ for the rational set of configurations C can be directly extended to finite TrPDS. This is a generalization of the saturation procedure for conditional pushdown systems [14]. A rational set of configurations of finite TrPDS is represented with automata that modify the rest of input by a transduction.

2 Preliminaries

2.1 Transducers

A *transducer* is a structure $(Q, \Gamma, \Delta, I, F)$ where Q is a finite set of states, Γ is a finite set of symbols, $\Delta \subseteq Q \times \Gamma^* \times \Gamma^* \times Q$ is a finite set of transition rules, $I \subseteq Q$ is the set of initial states, and $F \subseteq Q$ is the set of final states.

A computation of a transducer is a sequence of transitions of the following form:

$$p_0 \xrightarrow{w_1/w_1'} p_1 \xrightarrow{w_2/w_2'} \cdots \xrightarrow{w_n/w_n'} p_n$$

where $\langle p_{i-1}, w_i, w_i', p_i\rangle \in \Delta$. When we have the computation above, we write $p_0 \xrightarrow{w_1 \cdots w_n/w_1' \cdots w_n'} p_n$. The language $L(\mathfrak{t}) \subseteq \Gamma^* \times \Gamma^*$ of a transducer $\mathfrak{t}$ is defined as follows:

$$L(\mathfrak{t}) = \{\langle w_1, w_2\rangle \mid q_I \xrightarrow{w_1/w_2} q_f \text{ for some } q_I \in I \text{ and } q_F \in F\}$$

A *letter-to-letter transducer* is a transducer where Δ is restricted to $\Delta \subseteq Q \times \Gamma \times \Gamma \times Q$, *i.e.*, a letter-to-letter transducer is an automaton over $\Gamma \times \Gamma$.

2.2 Transductions

A *transduction* τ over Γ^* is a relation between Γ^* and Γ^*, or a function from Γ^* to $\mathcal{P}(\Gamma^*)$. A transduction τ is a *rational* transduction if there is a transducer $\mathfrak{t}$ such that $\tau = L(\mathfrak{t})$. A transduction τ is a *length-preserving* rational transduction if there is a *letter-to-letter transducer* $\mathfrak{t}$ such that $\tau = L(\mathfrak{t})$.

We write $\mathbf{Transd}_{\Gamma}$ for the set of all length-preserving rational transductions and use a metavariable $\mathfrak{t}$ to denote both a transducer and a transduction if there is no danger of confusion. In the rest of this paper, we use the term *transduction* for a length-preserving rational transduction.

The composition of transductions $\mathfrak{t}_1, \mathfrak{t}_2 \in \mathbf{Transd}_{\Gamma}$ is defined as that for relations:

$$\mathfrak{t}_1 \circ \mathfrak{t}_2 = \{\langle w, w'' \rangle \mid \langle w, w' \rangle \in \mathfrak{t}_1, \langle w', w'' \rangle \in \mathfrak{t}_2\}$$

$\mathbf{Transd}_{\Gamma}$ is closed under composition and $(\mathbf{Transd}_{\Gamma}, \circ, \cup, \mathbb{1}, \mathbb{0})$ is a semiring where $\mathbb{1} = \{\langle w, w \rangle \mid w \in \Gamma^*\}$ and $\mathbb{0} = \varnothing$.

By considering $\mathfrak{t} \in \mathbf{Transd}_{\Gamma}$ as a rational language over $\Gamma \times \Gamma$, we introduce (left) quotient and nullability $\nu(\mathfrak{t})$ defined as follows:

$$\begin{array}{ll}
\langle \cdot, \cdot \rangle^{-1} & : \forall w_1 \in \Gamma^*, w_2 \in \Gamma^{|w_1|}. \mathbf{Transd}_{\Gamma} \to \mathbf{Transd}_{\Gamma} \\
\langle \varepsilon, \varepsilon \rangle^{-1}\mathfrak{t} & = \mathfrak{t} \\
\langle \gamma, \gamma' \rangle^{-1}\mathfrak{t} & = \{\langle w, w' \rangle \mid \langle \gamma w, \gamma' w' \rangle \in \mathfrak{t}\} \\
\langle \gamma w, \gamma' w' \rangle^{-1}\mathfrak{t} & = \langle w, w' \rangle^{-1}(\langle \gamma, \gamma' \rangle^{-1}\mathfrak{t})
\end{array}$$

$$\nu(\mathfrak{t}) = \begin{cases} \{\varepsilon\} & \langle \varepsilon, \varepsilon \rangle \in \mathfrak{t} \\ \varnothing & \text{otherwise} \end{cases}$$

where $\varepsilon \in \Gamma^*$ is the empty string. $\mathbf{Transd}_{\Gamma}$ is closed under quotient and quotient distributes over composition in the following sense.

Proposition 1. $\langle w_1, w_2 \rangle^{-1}(\mathfrak{t}_1 \circ \mathfrak{t}_2) = \bigcup_{w_3 \in \Gamma^{|w_1|}} \left(\langle w_1, w_3 \rangle^{-1}\mathfrak{t}_1 \circ \langle w_3, w_2 \rangle^{-1}\mathfrak{t}_2 \right)$

A transduction $\mathfrak{t} \in \mathbf{Transd}_{\Gamma}$ can be considered as a function from Γ^* to $\mathcal{P}(\Gamma^*)$. We call this function application *action*, use the postfix notation, and write $w\mathfrak{t}$: $w\mathfrak{t} = \{w' \mid w' \in \Gamma^*, \langle w, w' \rangle \in \mathfrak{t}\}$.

The action is defined for a language L by $L\mathfrak{t} = \bigcup_{w \in L} w\mathfrak{t}$.

We can also inductively define action by using quotient:

$$\varepsilon\mathfrak{t} = \nu(\mathfrak{t}) \qquad (\gamma w)\mathfrak{t} = \bigcup_{\gamma' \in \Gamma} \left(\gamma' \lhd \left(w \, \langle \gamma, \gamma' \rangle^{-1}\mathfrak{t} \right) \right)$$

where $w \lhd W = \{ww' \mid w' \in W\}$.

2.3 Pushdown Systems

A pushdown system (PDS) is a structure $\mathcal{P} = (Q, \Gamma, \Delta)$ where Q is a finite set of control locations, Γ is a finite set of stack symbols, and $\Delta \subseteq Q \times (\Gamma^+ \times \Gamma^*) \times Q$ is a finite set of transition rules. For a transition rule $\langle p, \langle w_1, w_2 \rangle, q \rangle \in \Delta$, we write $p \xhookrightarrow{w_1/w_2} q$. A configuration of PDS $\mathcal{P}$ is a pair $\langle q, w \rangle$ of location $q \in Q$ and string $w \in \Gamma^*$. We write $\mathit{Conf}(\mathcal{P})$ for the set of all configurations $Q \times \Gamma^*$.

We define transition relation $\Rightarrow \subseteq Conf(\mathcal{P}) \times Conf(\mathcal{P})$: $\langle p, w_1 w\rangle \Rightarrow \langle q, w_2 w\rangle$ if $p \xhookrightarrow{w_1/w_2} q$ and $w \in \Gamma^*$.

We say a PDS $\mathcal{P} = (Q, \Gamma, \Delta)$ is a *standard* PDS if $|w| = 1$ for all $p \xhookrightarrow{w/w'} q \in \Delta$. It is clear that for a given PDS $\mathcal{P}$ we can construct a standard PDS equivalent to $\mathcal{P}$ by introducing extra states. We use a nonstandard PDS to simplify the construction of PDS for simulating TrPDS.

3 TrPDS : Pushdown Systems with Transductions

TrPDS is an extension of PDS that may modify the whole stack contents by applying a transduction.

A TrPDS is a structure (Q, Γ, T, Δ) where Q is a finite set of control locations, Γ is a finite set of stack symbols, $\Delta \subseteq Q \times (\Gamma \times \Gamma^* \times T) \times Q$ is a finite set of transition rules, and $T \subseteq \mathbf{Transd}_\Gamma$ is a finite set of transductions. For a transition rule $\langle p, \langle \gamma, w, \mathfrak{t}\rangle, q\rangle \in \Delta$, we write $p \xhookrightarrow{\gamma/w|\mathfrak{t}} q$ and call the triple "$\gamma/w|\mathfrak{t}$" *stack effect.*

A configuration of TrPDS $\mathcal{P}$ is a pair $\langle q, w\rangle$ of location $q \in Q$ and string $w \in \Gamma^*$. We write $Conf(\mathcal{P})$ for the set of all configurations $Q \times \Gamma^*$.

Definition 1 (Labelled transition relation). *We define a labelled transition relation* $\overset{\delta}{\Rightarrow} \subseteq Conf(\mathcal{P}) \times Conf(\mathcal{P})$*:* $\langle p, \gamma w'\rangle \overset{\delta}{\Rightarrow} \langle q, ww''\rangle$ *if* δ *is* $p \xhookrightarrow{\gamma/w|\mathfrak{t}} q$ *and* $w'' \in w'\mathfrak{t}$*. We also write* $c_1 \Rightarrow c_2$ *if* $c_1 \overset{\delta}{\Rightarrow} c_2$ *for some* $\delta \in \Delta$*.*

Let us consider an example of a TrPDS and its transitions.

Example 1. Let $\mathfrak{t}$ be $\langle b, b\rangle^* (\langle a, a\rangle \cup \langle a, b\rangle)^*$. TrPDS $\mathcal{P} = (Q, \Sigma, \{\mathfrak{t}, \mathbb{1}\}, \{\delta_1, \delta_2\})$ where $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $\delta_1 = q_0 \xhookrightarrow{a/\varepsilon|\mathfrak{t}} q_1$ and $\delta_2 = q_1 \xhookrightarrow{b/\varepsilon|\mathbb{1}} q_2$. The following are some examples of transitions.

$$\langle q_0, aaa\rangle \overset{\delta_1}{\Rightarrow} \langle q_1, ba\rangle \overset{\delta_2}{\Rightarrow} \langle q_2, a\rangle$$
$$\langle q_0, aaa\rangle \overset{\delta_1}{\Rightarrow} \langle q_1, bb\rangle \overset{\delta_2}{\Rightarrow} \langle q_2, b\rangle$$

The effect of transition rule $\delta = p \overset{\sigma}{\hookrightarrow} q$ with stack effect σ is captured by the following function effect_σ below: $\langle p, w\rangle \overset{\delta}{\Rightarrow} \langle q, w'\rangle$ iff $w' \in \mathit{effect}_\sigma(w)$.

$$\begin{array}{ll} \mathit{effect}_{\gamma/w|\mathfrak{t}} & : \Gamma^* \to \mathcal{P}(\Gamma^*) \\ \mathit{effect}_{\gamma/w|\mathfrak{t}}(\varepsilon) & = \varnothing \\ \mathit{effect}_{\gamma/w|\mathfrak{t}}(\gamma' w') & = \begin{cases} w \lhd w'\mathfrak{t} & \text{if } \gamma' = \gamma \\ \varnothing & \text{otherwise} \end{cases} \end{array}$$

Definition 2. *The closure* $\langle T\rangle$ *of a transduction set* T *under composition and quotient is inductively defined as follows.*

- $T \subseteq \langle T\rangle$ *and* $\mathbb{0}, \mathbb{1} \in \langle T\rangle$*.*
- *If* $\mathfrak{t}_1, \mathfrak{t}_2 \in \langle T\rangle$*, then* $\mathfrak{t}_1 \circ \mathfrak{t}_2 \in \langle T\rangle$*.*
- *If* $\mathfrak{t} \in \langle T\rangle$*, then* $\langle \gamma, \gamma'\rangle^{-1}\mathfrak{t} \in \langle T\rangle$ *for all* $\gamma, \gamma' \in \Gamma$*.*

Definition 3. *A TrPDS $\mathcal{P}$ over T is called* finite *if $\langle T\rangle$ is finite.*

Example: Conditional Pushdown Systems. A conditional pushdown system is a pushdown system extended with stack inspection [9,12]. A transition rule has the form $\langle p, \langle\gamma, w, L\rangle, q\rangle \in \Delta$ where L is a rational language over stack symbols[1]: it induces the transition relation $\langle p, \gamma w'\rangle \Rightarrow \langle q, ww'\rangle$ if $w' \in L$. The transition can be taken only when $w' \in L$.

Let $\mathcal{L}$ be the finite set of rational languages appearing in transition rules. Here, we define $\langle\mathcal{L}\rangle$ inductively as follows:

- $\mathcal{L} \subseteq \langle\mathcal{L}\rangle$ and $\varnothing, \Gamma^* \in \langle\mathcal{L}\rangle$.
- If $L_1, L_2 \in \langle\mathcal{L}\rangle$, then $L_1 \cap L_2 \in \langle\mathcal{L}\rangle$.
- If $L \in \langle\mathcal{L}\rangle$, then $\gamma^{-1}L \in \langle\mathcal{L}\rangle$ for all $\gamma \in \Gamma$.

The set $\langle\mathcal{L}\rangle$ is finite since quotient distributes over intersection and there are finitely many languages obtained from each rational language with quotient.

For a language L, we define $\widetilde{L} = \{\langle w, w\rangle \mid w \in L\}$ and then $\widetilde{L}$ is a length-preserving rational transduction for a rational language L. For the composition and the quotient on $\widetilde{L}$, we have the following.

$$\widetilde{L} \circ \widetilde{L'} = \widetilde{L \cap L'} \quad \langle\gamma, \gamma'\rangle^{-1}\widetilde{L} = \begin{cases} \widetilde{\gamma^{-1}L} & \text{if } \gamma = \gamma' \\ \mathbb{0} & \text{otherwise} \end{cases}$$

Then, a conditional pushdown system over $\mathcal{L}$ can be considered as a *finite* TrPDS over the transduction set $T = \{\widetilde{L} \mid L \in \mathcal{L}\}$. It is clear that $\langle T\rangle$ is finite since we have $\langle T\rangle = \{\widetilde{L} \mid L \in \langle\mathcal{L}\rangle\}$ from the properties above.

Example: Transformable Pushdown Systems. A transformable pushdown system is a pushdown system that may modify stack by applying a function over stack symbols. This generalizes the operation of discrete timed pushdown systems [2] that increment the ages of stack symbols. A transition rule has the form $\langle p, \langle\gamma, w, f\rangle, q\rangle \in \Delta$ where f is a function from Γ to Γ: it induces the transition relation $\langle p, \gamma w'\rangle \Rightarrow \langle q, wf(w')\rangle$.

For a given $f \in \Gamma \to \Gamma$, we define a transduction $\widehat{f} = \{\langle w, f(w)\rangle \mid w \in \Gamma^*\}$. It is clear that $\widehat{f}$ is a length-preserving rational transduction and the following hold.

$$\widehat{f_1} \circ \widehat{f_2} = \widehat{f_2 \circ f_1} \quad \langle\gamma, \gamma'\rangle^{-1}\widehat{f} = \begin{cases} \widehat{f} & \text{if } \gamma' = f(\gamma) \\ \mathbb{0} & \text{otherwise} \end{cases}$$

Note that $\circ$ in $f_2 \circ f_1$ is the composition of functions, *i.e.*, $f_2 \circ f_1 = \lambda v. f_2(f_1(v))$.

A transformable pushdown system over $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$ can be considered as a *finite* TrPDS over $T = \{\widehat{f_1}, \widehat{f_2}, \ldots, \widehat{f_n}\}$. It is clear that $\langle T\rangle$ is finite because $\langle T\rangle \subseteq \{\widehat{f} \mid f \in \Gamma \to \Gamma\} \cup \{\mathbb{0}\}$.

---

[1] We use the definitions of transition rule and transition relation inspecting whole stack without its top [12] rather than inspecting whole stack [9].

Example: Two-counter Machines. Any two-counter machine without input can be simulated by a TrPDS $\mathcal{P} = (Q, \{0, 1, \lambda\}, \{\mathfrak{d}_0, \mathfrak{t}_0, \mathfrak{d}_1, \mathfrak{t}_1\}, \Delta)$. The transduction $\mathfrak{d}_0$ decrements the number of 0's in the stack by replacing the first 0 with λ and the transduction $\mathfrak{t}_0$ checks whether stack contains 0 or not. $\mathfrak{d}_1$ and $\mathfrak{t}_1$ have the same behaviors for 1's.

Since $\mathfrak{d}_0 \neq \mathfrak{d}_0^2 \neq \cdots \neq \mathfrak{d}_0^n \neq \cdots$, $\mathcal{P}$ is not a finite TrPDS. The reachability problem of two-counter machines is undecidable and thus the reachability problem of TrPDS is undecidable in general. However, we will show that the reachability problem of a *finite* TrPDS is decidable.

4 Construction of PDS from TrPDS

From a given finite TrPDS, we construct a finite PDS by lazily applying a transduction to stack. It is a generalization of the construction introduced by Abdulla et al. [2] for simulating discrete timed pushdown systems. By the construction, we prove that the reachability problem of a finite TrPDS is decidable.

4.1 Construction

For a given TrPDS $\mathcal{P} = (Q, \Gamma, T, \Delta)$, we construct a PDS $\mathcal{P}' = (Q, \Gamma \cup \langle T \rangle, \Delta')$.

Let δ be a transition rule $p \xhookrightarrow{\gamma/w|\mathfrak{t}} q \in \Delta$. We have transition $\langle p, \gamma w' \rangle \overset{\delta}{\Rightarrow} \langle q, ww'' \rangle$ in $\mathcal{P}$ for $w'' \in w'\mathfrak{t}$ where the transduction $\mathfrak{t}$ is applied to the rest of stack w'. In ordinary pushdown systems, we can only modify the top of stack at each transition. Thus, we delay the application of the transduction in $\mathcal{P}'$ by keeping it on stack.

We construct three kinds of transition rules of $\mathcal{P}'$ as follows:

$$
\begin{array}{lll}
\mathit{APPLY}\text{-transition} & : p \xhookrightarrow{\gamma/w\,\mathfrak{t}} q \in \Delta' & \text{if } p \xhookrightarrow{\gamma/w|\mathfrak{t}} q \in \Delta \\
\mathit{COMPOSE}\text{-transition} & : p \xhookrightarrow{\mathfrak{t}_1\mathfrak{t}_2/\mathfrak{t}_2 \circ \mathfrak{t}_1} p \in \Delta' & \text{if } \mathfrak{t}_1, \mathfrak{t}_2 \in \langle T \rangle \\
\mathit{UNFOLD}\text{-transition} & : p \xhookrightarrow{\mathfrak{t}\gamma/\gamma' \langle \gamma, \gamma' \rangle^{-1}\mathfrak{t}} p \in \Delta' & \text{if } \mathfrak{t} \in \langle T \rangle \text{ and } \gamma, \gamma' \in \Gamma
\end{array}
$$

With *APPLY*-transition rule, we have transition $\langle p, \gamma w' \rangle \Rightarrow \langle q, w\mathfrak{t}w' \rangle$ in $\mathcal{P}'$. The application of transduction $\mathfrak{t}$ is simulated lazily with *COMPOSE*-transition rule and *UNFOLD*-transition rule.

For $\delta \in \Delta$, we have a corresponding *APPLY*-transition rule in Δ' and we write $\overset{\delta}{\Rightarrow}$ for a transition relation obtained from δ. Similarly, we write $\overset{C}{\Rightarrow}$ and $\overset{U}{\Rightarrow}$ for transition relations induced by *COMPOSE*-transition rule and *UNFOLD*-transition rule, respectively. We also write $\langle p, w \rangle \Rightarrow \langle q, w' \rangle$ if $\langle p, w \rangle \overset{\delta}{\Rightarrow} \langle q, w' \rangle$ for some δ, $\langle p, w \rangle \overset{C}{\Rightarrow} \langle q, w' \rangle$, or $\langle p, w \rangle \overset{U}{\Rightarrow} \langle q, w' \rangle$.

For $\delta \in \Delta$, we introduce many-steps transition relation $\overset{\delta}{\Rrightarrow} = \overset{C}{\Rightarrow}^* \circ \overset{U}{\Rightarrow}^? \circ \overset{\delta}{\Rightarrow}$ in $\mathcal{P}'$ ($\circ$ is used as binary relation composition).

Example 2. The following are the transitions of the constructed PDS that correspond to those in Example 1.

$$\langle q_0, aaa\rangle \overset{\delta_1}{\Rrightarrow} \langle q_0, \mathfrak{t}aa\rangle \overset{U}{\Rrightarrow} \langle q_0, b\mathfrak{t}'a\rangle \overset{\delta_2}{\Rrightarrow} \langle q_1, \mathbb{1}\mathfrak{t}'a\rangle \overset{C}{\Rrightarrow} \langle q_1, \mathfrak{t}'a\rangle \overset{U}{\Rrightarrow} \langle q_2, a\,\langle a, a\rangle^{-1}\mathfrak{t}'\rangle$$
$$\langle q_0, aaa\rangle \overset{\delta_1}{\Rrightarrow} \langle q_0, \mathfrak{t}aa\rangle \overset{U}{\Rrightarrow} \langle q_0, b\mathfrak{t}'a\rangle \overset{\delta_2}{\Rrightarrow} \langle q_1, \mathbb{1}\mathfrak{t}'a\rangle \overset{C}{\Rrightarrow} \langle q_1, \mathfrak{t}'a\rangle \overset{U}{\Rrightarrow} \langle q_2, b\,\langle a, b\rangle^{-1}\mathfrak{t}'\rangle$$

(For the sake of simplicity, we abbreviate $\langle a, b\rangle^{-1}\mathfrak{t}$ as $\mathfrak{t}'$.)

For $\delta = p \overset{\sigma}{\hookrightarrow} q \in \Delta$, we have $\langle p, w\rangle \overset{\delta}{\Rrightarrow} \langle q, w'\rangle$ iff $w' \in \mathit{Effect}_\sigma(w)$ where *Effect* is inductively defined as follows.

$$\begin{array}{ll}
\mathit{Effect}_{\gamma/w|\mathfrak{t}} & : (\Gamma \cup \langle T\rangle)^* \to \mathcal{P}((\Gamma \cup \langle T\rangle)^*) \\
\mathit{Effect}_{\gamma/w|\mathfrak{t}}(\varepsilon) & = \varnothing \\
\mathit{Effect}_{\gamma/w|\mathfrak{t}}(\mathfrak{t}') & = \varnothing \\
\mathit{Effect}_{\gamma/w|\mathfrak{t}}(\gamma' w') & = \begin{cases} \{w\,\mathfrak{t}\,w'\} & \text{if } \gamma = \gamma' \\ \varnothing & \text{otherwise} \end{cases} \\
\mathit{Effect}_{\gamma/w|\mathfrak{t}}(\mathfrak{t}'\gamma' w') & = \{w\,\mathfrak{t}\,(\langle \gamma', \gamma\rangle^{-1}\mathfrak{t}')\,w'\} \\
\mathit{Effect}_{\gamma/w|\mathfrak{t}}(\mathfrak{t}_1\mathfrak{t}_2 w') & = \mathit{Effect}_{\gamma/w|\mathfrak{t}}((\mathfrak{t}_2 \circ \mathfrak{t}_1) w')
\end{array}$$

4.2 Simulation

To reveal a relation between TrPDS and PDS, we consider the difference in how a transduction is applied to stacks of a TrPDS and a PDS. For a TrPDS, a transduction is applied to its stack immediately. On the other hand, for the PDS constructed from the TrPDS, the corresponding transduction is lazily applied to its stack when the PDS takes transitions that unfold transductions. This difference is reflected in the definitions of $\mathit{effect}_\sigma(w)$ and $\mathit{Effect}_\sigma(w)$.

To relate stacks of a TrPDS and a PDS, we introduce *concretization* to obtain the set of stacks of the TrPDS from a stack of the PDS.

Definition 4 (concretization of stack).

$$\begin{array}{ll}
\|\cdot\| & : (\Gamma \cup \langle T\rangle)^* \to \mathcal{P}(\Gamma^*) \\
\|\varepsilon\| & = \{\varepsilon\} \\
\|\gamma w\| & = \gamma \lhd \|w\| \\
\|\mathfrak{t} w\| & = \|w\|\mathfrak{t}
\end{array}$$

The gap between $\mathit{effect}_\sigma(w)$ and $\mathit{Effect}_\sigma(w)$ is filled by applying the concretization of stack.

Lemma 1. *For all stack effect σ, $\|\mathit{Effect}_\sigma(w)\| = \mathit{effect}_\sigma(\|w\|)$*

To establish the simulation, we consider transitions of a set of configurations and define $\mathit{post}_\delta(C)$ for a TrPDS and $\mathit{Post}_\delta(C)$ for a constructed PDS.

$$\mathit{post}_\delta(C) = \{c' \mid c \in C, c \underset{\mathcal{P}}{\overset{\delta}{\Rightarrow}} c'\} \quad \mathit{Post}_\delta(C) = \{c' \mid c \in C, c \underset{\mathcal{P}'}{\overset{\delta}{\Rrightarrow}} c'\}$$

Transitions of a set of configurations can be related by extending concretization for configurations.

$$\| \langle p, w \rangle \| = \{ \langle p, w' \rangle \mid w' \in \|w\| \} \qquad \|C\| = \bigcup_{c \in C} \|c\|$$

Theorem 1. *For any set* $C \subseteq Q \times (\Gamma \cup \langle T \rangle)^*$, $post_\delta(\|C\|) = \|Post_\delta(C)\|$.

Proof. Let δ be a transition $p \overset{\sigma}{\hookrightarrow} q$.

$$\begin{aligned}
\|Post_\delta(C)\| &= \|\{c' \mid c \in C, c \overset{\delta}{\Rrightarrow} c'\}\| \\
&= \|\{\langle q, w' \rangle \mid \langle p, w \rangle \in C, w' \in Effect_\sigma(w)\}\| \\
&= \{\langle q, w' \rangle \mid \langle p, w \rangle \in C \ , w' \in \|Effect_\sigma(w)\|\} \\
&= \{\langle q, w' \rangle \mid \langle p, w \rangle \in C \ , w' \in effect_\sigma(\|w\|)\} \ \text{(Lemma 1)} \\
&= \{\langle q, w' \rangle \mid \langle p, w \rangle \in \|C\|, w' \in effect_\sigma(w)\} \\
&= \{c' \mid c \in \|C\|, c \overset{\delta}{\Rightarrow} c'\} = post_\delta(\|C\|)
\end{aligned}$$

□

We then consider one-step transitions of a set of configurations that may apply any transition rule.

$$post(C) = \bigcup_{\delta \in \Delta} post_\delta(C) \qquad Post(C) = \{c' \mid c \in C, c \underset{\mathcal{P}'}{\Rrightarrow} c'\}$$

It should be noted that the definition of $Post(C)$ does not directly correspond to that of $Post_\delta(C)$ because $Post(C)$ captures one-step transitions while $Post_\delta(C)$ captures many-steps transitions. However, we have the following weaker correspondence.

$$Post_\delta(C) \subseteq Post^*(C) \qquad \|Post(C)\| \subseteq \|C \cup \bigcup_{\delta \in \Delta} Post_\delta(C)\|$$

From Theorem 1, for any set $C \subseteq Q \times \Gamma^*$, we have $post_\delta(C) = \|Post_\delta(C)\|$ and obtain the following corollary.

Corollary 1. *For any set* $C \subseteq Q \times \Gamma^*$, $post^*(C) = \|Post^*(C)\|$.

4.3 Computing *Post**

In this section, we show the forward reachable set $post^*(C)$ for a rational set of configurations C is rational and effectively computable. Thus, the reachability problem of a finite TrPDS is decidable.

To compute $post^*(C)$, we use Corollary 1 and apply usual (forward) reachability analysis to calculate $Post^*(C)$ [8]. For calculating the concretized set of configurations $\|Post^*(C)\|$, we introduce a tail recursive version of $\|\cdot\|$ as follows:

$$\begin{aligned}
\|\cdot\|' \quad &: (\Gamma \cup \langle T \rangle)^* \times \langle T \rangle \to \mathcal{P}(\Gamma^*) \\
\|\varepsilon\|'_{\mathfrak{a}} &= \nu(\mathfrak{a}) \\
\|\gamma w\|'_{\mathfrak{a}} &= \bigcup_{\gamma' \in \Gamma} \left(\gamma' \lhd \|w\|'_{\langle \gamma, \gamma' \rangle^{-1} \mathfrak{a}} \right) \\
\|\mathfrak{t} w\|'_{\mathfrak{a}} &= \|w\|'_{\mathfrak{t} \circ \mathfrak{a}} \\
\|w\|' &= \|w\|'_{\mathbb{1}}
\end{aligned}$$

We prove the equivalence of the two versions by induction on w.

Proposition 2. $\|w\|\mathfrak{t} = \|w\|'_{\mathfrak{t}}$.

It should be noted that the function $\|\cdot\|'$ is realized as a transducer [2]. The key of the construction is to consider accumulator $\mathfrak{a}$ as a state of the transducer.

To be exact, we construct the transducer $\mathfrak{c} = (\langle T\rangle, \Gamma, \Delta, I, F)$ where $I = \{\mathbb{1}\}$ and $F = \{\mathfrak{t} \mid \mathfrak{t} \in \langle T\rangle, \nu(\mathfrak{t}) = \{\varepsilon\}\}$.

$$\begin{array}{ll} \mathfrak{a} \xrightarrow{\gamma/\gamma'} (\langle\gamma,\gamma'\rangle^{-1}\mathfrak{a}) \in \Delta & \quad \text{for all } \gamma' \in \Gamma \\ \mathfrak{a} \xrightarrow{\mathfrak{t}/\varepsilon} (\mathfrak{t} \circ \mathfrak{a}) \in \Delta & \quad \text{for all } \mathfrak{t} \in \langle T\rangle \end{array}$$

Then, we have $\|w\|' = w\mathfrak{c}$.

For a rational set of configurations C, $Post^*(C)$ is rational from forward reachability analysis. Thus, we can effectively compute $(Post^*(C))\mathfrak{c}$ since it is the image under transducer $\mathfrak{c}$ of rational set $Post^*(C)$.

Finally, we obtain the following theorem.

Theorem 2. *For a rational set of configurations C of a finite TrPDS, post$^*(C)$ is rational and effectively computable.*

5 Conditional Transformable Pushdown Systems

We consider conditional transformable pushdown systems as a nontrivial subclass of finite TrPDS. Such pushdown systems may have both kinds of transition rules of conditional and transformable pushdown systems.

A conditional transformable pushdown system is a TrPDS $(Q, \Gamma, \widetilde{\mathcal{L}} \cup \widehat{\mathcal{F}}, \Delta)$ where $\mathcal{L}$ is a finite set of rational languages over Γ and $\mathcal{F}$ is a finite set of functions over Γ. We show that $\langle\widetilde{\mathcal{L}} \cup \widehat{\mathcal{F}}\rangle$ is finite and thus any conditional transformable pushdown system is a finite TrPDS. Hence, the reachability problem of conditional transformable pushdown systems is decidable.

In order to show that $\langle\widetilde{\mathcal{L}} \cup \widehat{\mathcal{F}}\rangle$ is finite, we introduce a notion of *implementation*. We define an algebra $(U, \bullet, \langle\cdot,\cdot\rangle^{-1})$ which is closed under composition and quotient as follows:

- $\bullet : U \times U \to U$ is a binary operator that corresponds to composition, and
- $\langle\cdot,\cdot\rangle^{-1} : \Gamma \times \Gamma \times U \to U$ is a ternary operator that corresponds to quotient.

Then, we define an *implementation* of a finite transduction set T.

Definition 5 (Implementation of T). *For a given finite transduction set T, we call an algebra $(U, \bullet, \langle\cdot,\cdot\rangle^{-1})$ equipped with functions $F : T \to U$ and $G : U \to$* **Transd**$_\Gamma$ *an* implementation *of T if the following hold:*

- $G \circ F = id$
- $G(\mathfrak{u}_1 \bullet \mathfrak{u}_2) = G(\mathfrak{u}_1) \circ G(\mathfrak{u}_2)$

[2] This transducer is not letter-to-letter.

– $G(\langle\gamma,\gamma'\rangle^{-1}\mathfrak{u}) = \langle\gamma,\gamma'\rangle^{-1}G(\mathfrak{u})$

We use the following proposition to show that $\langle T\rangle$ is finite.

Proposition 3. *For a given finite transduction set T, $\langle T\rangle$ is finite if there is a finite implementation of T.*

The following property is the key to the construction of a finite implementation of $\widetilde{\mathcal{L}} \cup \widehat{\mathcal{F}}$.

Proposition 4. *Let $L \subseteq \Gamma^*$ and $h : \Gamma \to \Gamma$. Then, we have $\widehat{h} \circ \widetilde{L} = \widetilde{h^{-1}(L)} \circ \widehat{h}$.*

This property implies any sequence $\widehat{h_1} \circ \widetilde{L_1} \circ \widehat{h_2} \circ \widetilde{L_2} \circ \cdots \circ \widehat{h_i} \circ \widetilde{L_i}$ can be normalized as $(\widetilde{L'_1} \circ \widetilde{L'_2} \circ \cdots \circ \widetilde{L'_i}) \circ (\widehat{h_1} \circ \widehat{h_2} \circ \cdots \circ \widehat{h_i})$. It means that the inspection of the stack can be done before modification.

Based on this property, we define an implementation $\mathcal{I}_T = (\mathcal{C} \times \langle\widehat{\mathcal{F}}\rangle, \bullet, \langle\cdot,\cdot\rangle^{-1})$ with F and G where $\mathcal{C}$ is inductively defined as follows:

- $\mathcal{L} \subseteq \mathcal{C}$ and $\varnothing, \Gamma^* \in \mathcal{C}$.
- If $L_1, L_2 \in \mathcal{C}$, then $L_1 \cap L_2 \in \mathcal{C}$.
- If $L \in \mathcal{C}$ and $\gamma \in \Gamma$, then $\gamma^{-1}L \in \mathcal{C}$.
- If $L \in \mathcal{C}$ and $h \in \mathcal{F}$, then $h^{-1}L \in \mathcal{C}$.

The set $\mathcal{C}$ is finite because $\gamma^{-1}(h^{-1}L) = h^{-1}((h(\gamma))^{-1}L)$, $h^{-1}(g^{-1}(L)) = (g \circ h)^{-1}(L)$, and $h^{-1}(L_1 \cap L_2) = h^{-1}L_1 \cap h^{-1}L_2$.

We define the operators and functions of the implementation as follows:

$$\begin{aligned}
\langle L_1, h_1\rangle \bullet \langle L_2, h_2\rangle &= \left\langle L_1 \cap h_1^{-1}(L_2), h_1 \circ h_2\right\rangle \\
\langle\gamma,\gamma'\rangle^{-1}\langle L, h\rangle &= \left\langle \gamma^{-1}L, \langle\gamma,\gamma'\rangle^{-1}h\right\rangle \\
F(\mathfrak{t}) &= \begin{cases} \langle L, \mathbb{1}\rangle & \text{if } \mathfrak{t} \in \widetilde{\mathcal{L}} \text{ and } \mathfrak{t} = \widetilde{L} \\ \langle \Gamma^*, \mathfrak{t}\rangle & \text{otherwise} \end{cases} \\
G(\langle L, h\rangle) &= \widetilde{L} \circ h
\end{aligned}$$

With respect to F and G, we need to show that the three conditions of implementations hold: $G \circ F = id$ and $G(\langle\gamma,\gamma'\rangle^{-1}\mathfrak{u}) = \langle\gamma,\gamma'\rangle^{-1}G(\mathfrak{u})$ are easily proved from the definition and we use Proposition 4 to prove $G(\mathfrak{u}_1 \bullet \mathfrak{u}_2) = G(\mathfrak{u}_1) \circ G(\mathfrak{u}_2)$.

Finally, we obtain the following corollary of Theorem 2.

Corollary 2. *For a rational set of configurations C of a conditional transformable pushdown system, $post^*(C)$ is rational and effectively computable.*

6 Saturation Procedure of TrPDS

We extend the saturation procedure of PDS for finite TrPDS which computes the set $pre^*(C)$ backward reachable from a rational set of configurations C [5,10].

First, we review the saturation procedure for ordinary pushdown systems. Then, we extend the saturation procedure for TrPDS based on that for conditional pushdown systems where a rational set of configurations is represented by an automaton with regular lookahead [14]. In particular, we introduce automata with transductions (TrNFA) that apply transductions to the rest of the input and extend the saturation procedure so that it constructs a TrNFA from a given finite TrPDS.

6.1 Saturation Procedure for PDS

We review the ordinary saturation procedure. To simplify our presentation, we first consider the set of configurations backward reachable from a single configuration $\langle q_f, \varepsilon\rangle$. For a given PDS $\mathcal{P} = (Q, \Gamma, \Delta)$, we construct a finite automaton $\mathcal{A}_\omega$ that accepts $pre^*(\langle q_f, \varepsilon\rangle)$ where $q_f \in Q$.

The saturation procedure starts from the initial $\mathcal{P}$-automaton $\mathcal{A}_0$ and iteratively updates $\mathcal{P}$-automaton $\mathcal{A}_i$ into $\mathcal{A}_{i+1}$ until saturation. The saturation procedure is described as follows:

1. Let the initial $\mathcal{P}$-automaton $\mathcal{A}_0$ be $(Q, \Gamma, \varnothing, Q, \{q_f\})$.
2. If $p \xhookrightarrow{\gamma/w} q \in \Delta$ and $q \xrightarrow[\mathcal{A}_i]{w} p'$, then we obtain $\mathcal{A}_{i+1}$ by adding transition $\langle p, \gamma, p'\rangle$ to $\mathcal{A}_i$.
3. Repeat 2 until saturation.

This procedure always terminates since $Q \times \Gamma \times Q$ is finite, and we obtain a fixed point $\mathcal{P}$-automaton $\mathcal{A}_\omega$.

The constructed $\mathcal{P}$-automaton $\mathcal{A}_\omega$ has the following property and hence we have $L(\mathcal{A}_\omega) = pre^*(\langle q_f, \varepsilon\rangle)$.

Theorem 3. *$p \xrightarrow[\mathcal{A}_\omega]{w}{}^* q$ iff $\langle p, w\rangle \Rightarrow \langle q, \varepsilon\rangle$.*

The saturation procedure above can be used to compute $pre^*(C)$ for a rational set of configurations C. Let $C \subseteq Q \times \Gamma^*$ be a rational set of configurations accepted by $\mathcal{P}$-automaton $B = (P, \Gamma, \Delta', Q, F)$. Without loss of generality, we can assume B has no transition leading to an initial state. We construct new PDS $\mathcal{P}' = (Q \cup P, \Gamma, \Delta'')$ where $\Delta'' = \Delta \cup \{p \xhookrightarrow{\gamma/\varepsilon} q \mid \langle p, \gamma, q\rangle \in \Delta'\}$.

Then, $pre^*(F \times \{\varepsilon\})$ in PDS $\mathcal{P}'$ is equal to $pre^*(C)$ in PDS $\mathcal{P}$. Hence, we only consider the set of configurations backward reachable from a single configuration with empty stack in the following sections.

6.2 TrNFA

To represent a rational set of configurations of finite TrPDS, we introduce automata with transductions (TrNFA) that apply transductions to the rest of the input.

A TrNFA is a structure $A = (Q, \Sigma, \Delta, T, I, F)$ where Q is a finite set of states, Σ is a finite set of symbols, $\Delta \subseteq Q \times (\Sigma \to \langle\!\langle T\rangle\!\rangle) \times Q$ is a finite set of transition rules, T is a finite set of transductions, $I \subseteq Q$ is a set of initial states, and $F \subseteq Q$ is a set of final states. $\langle\!\langle T\rangle\!\rangle$ is the smallest set S such that $\langle T\rangle \subseteq S$ and S is closed under union $\cup$.

We write $p \xrightarrow{\gamma|\mathfrak{t}} q$ if $\langle p, \sigma, q\rangle \in \Delta$ and $\sigma(\gamma) = \mathfrak{t}$. Transition $p \xrightarrow{\gamma|\mathfrak{t}} q$ means that the automaton consumes γ from input, transforms the rest of input by $\mathfrak{t}$, and changes its state from p to q.

Intuitively, the composition of two transitions could be defined as follows:

$$p \xrightarrow{\gamma\gamma''|\langle\gamma'',\gamma'\rangle^{-1}\mathfrak{t}\circ\mathfrak{t}'} q \qquad \text{if} \quad p \xrightarrow{\gamma|\mathfrak{t}} r,\ r \xrightarrow{\gamma'|\mathfrak{t}'} q, \text{ and } \gamma'' \in \Sigma$$

With the above definition, finitely *many* transitions accrue by the composing the two transitions, and then the associativity of the composition of transitions does not hold. On the other hand, we obtain *only one* transition by composing two transitions in usual automata and the associativity of the composition holds.

To deal with this problem, we define product $\otimes$ over $\Sigma^* \to \langle\langle T\rangle\rangle$ and introduce pseudo formal power series semiring $(\mathcal{S}, \otimes, \oplus, 1, 0)$.

Definition 6 (Pseudo formal power series semiring).

$$\mathcal{S} = \Sigma^* \to \langle\langle T\rangle\rangle, \qquad 1 = \lambda w. \begin{cases} \mathbb{1} & \textit{if } w = \varepsilon \\ \mathbb{0} & \textit{otherwise} \end{cases}, \qquad 0 = \lambda w.\mathbb{0}$$

$$(\sigma_1 \otimes \sigma_2)(w) = \bigcup_{\substack{w = w_1 w_3 \\ |w_3| = |w_2|}} \Big(\langle w_3, w_2\rangle^{-1} \sigma_1(w_1) \circ \sigma_2(w_2) \Big)$$
$$(\sigma_1 \oplus \sigma_2)(w) = \sigma_1(w) \cup \sigma_2(w)$$

We define inductively transition relations as follows:

$$\begin{array}{rll} p \xrightarrow{1} p & & \\ p \xrightarrow{\sigma} q & \text{if} & \langle p, \sigma, q\rangle \in \Delta \\ p \xrightarrow{\sigma_1 \otimes \sigma_2} r & \text{if} & p \xrightarrow{\sigma_1} q \text{ and } q \xrightarrow{\sigma_2} r \end{array}$$

The associativity of composition of transitions holds as a result of bundling transitions.

We write $p \xrightarrow{w|\mathfrak{t}} q$ if $p \xrightarrow{\sigma} q$ and $\sigma(w) = \mathfrak{t}$, and $p \xrightarrow{w} q$ if $p \xrightarrow{w|\mathfrak{t}} q$ and $\nu(\mathfrak{t}) = \{\varepsilon\}$. Even if $p \xrightarrow{w|\mathfrak{t}} q$, we have a transition from p to q consuming w only when the rest of input is successfully transformed by $\mathfrak{t}$. We define the language of automaton : $L(A) = \{w \mid p \xrightarrow{w} q \text{ for some } p \in I, q \in F\}$.

6.3 Computing *pre** of TrPDS

To compute $pre^*(\langle q_f, \varepsilon\rangle)$ of TrPDS $\mathcal{P} = (Q, \Gamma, \Delta, T)$, we start from the initial TrNFA $A_0 = (Q, \Gamma, \Delta', T, Q, \{q_f\})$ where $\Delta' = \{\langle p, \lambda\gamma.\mathbb{0}, q\rangle \mid p, q \in Q\}$.

To construct a TrNFA that accepts $pre^*(\langle q_f, \varepsilon\rangle)$, we extend the saturation rule as follows:

– If $p \xhookrightarrow{\gamma/w|\mathfrak{t}} q \in \Delta$, $q \xrightarrow{w|\mathfrak{t}'} p'$ in the current automaton, and $\langle p, \sigma, p' \rangle \in \Delta'$, then we replace $\langle p, \sigma, p' \rangle$ by $\langle p, \sigma \oplus \sigma', p' \rangle$ where $\sigma'(\gamma) = \mathfrak{t} \otimes \mathfrak{t}'$ and $\sigma'(\gamma') = \mathbb{0}$ if $\gamma \neq \gamma'$.

The saturation procedure always terminates and calculates the fixed point automaton A_ω because $\Sigma \to \langle\!\langle T \rangle\!\rangle$ is finite.

We have the following two lemmas that bridge a computation of TrPDS and a behavior of TrNFA.

Lemma 2. *If* $\langle p, w \rangle \Rightarrow^* \langle q, \varepsilon \rangle$, *then* $p \xrightarrow[A_\omega]{w} q$.

Lemma 3. *If* $p \xrightarrow[A_\omega]{w|\mathfrak{t}} q$, *then* $\langle p, ww' \rangle \Rightarrow^* \langle q, w'' \rangle$ *for all* $w'' \in w'\mathfrak{t}$.

From these lemmas, we have the following theorem that implies $L(A_\omega) = pre^*(\langle q_f, \varepsilon \rangle)$.

Theorem 4. $p \xrightarrow[A_\omega]{w} q$ *iff* $\langle p, w \rangle \Rightarrow^* \langle q, \varepsilon \rangle$.

6.4 Construction of Automata from TrNFA

We construct a finite automaton A' from a finite TrNFA $A = (S, \Sigma, \Delta, T, I, F)$ to show that pre^* is rational and effectively computable.

The construction is very simple. We construct the finite automaton $A' = (S \times \langle\!\langle T \rangle\!\rangle, \Sigma, \Delta', I \times \{\mathbb{1}\}, F \times \{\mathfrak{t} \mid \mathfrak{t} \in \langle\!\langle T \rangle\!\rangle, \nu(\mathfrak{t}) = \{\varepsilon\}\})$ where each state $p_\mathfrak{t}$ of A' means that we must apply $\mathfrak{t}$ to the rest of input. For each $p \xrightarrow{\gamma|\mathfrak{t}} q \in \Delta$, we add transition $\left\langle p_\mathfrak{u}, \gamma', q_{\langle \gamma', \gamma 1 \rangle^{-1} \mathfrak{u} \circ \mathfrak{t}} \right\rangle$ into Δ' for all $\gamma' \in \Gamma, \mathfrak{u} \in \langle\!\langle T \rangle\!\rangle$.

To distinguish transitions of finite automata from those of TrNFA, we write $p_\mathfrak{t} \overset{w}{\rightharpoonup} p'_{\mathfrak{t}'}$ for transitions of A'. From the definition of A', we have $L(A') = \{w \mid p_\mathbb{1} \overset{w}{\rightharpoonup} q_\mathfrak{t}$ and $\nu(\mathfrak{t}) = \{\varepsilon\}$ for some $p \in I$, $q \in F$, and $\mathfrak{t} \in \langle\!\langle T \rangle\!\rangle\}$.

Then, we have the following two lemmas: Lemma 4 states that the constructed automaton captures behaviors of TrNFA and Lemma 5 states the other direction.

Lemma 4. *If* $p \xrightarrow{w|\mathfrak{t}} q$, *then there exist* $\mathfrak{t}_1, \mathfrak{t}_2, \ldots, \mathfrak{t}_n$ *such that* $p_\mathbb{1} \overset{w}{\rightharpoonup} q_{\mathfrak{t}_1}, \ldots, p_\mathbb{1} \overset{w}{\rightharpoonup} q_{\mathfrak{t}_n}$ *and* $\mathfrak{t} = \bigcup_{1 \leq i \leq n} \mathfrak{t}_i$.

Lemma 5. *If* $p_\mathbb{1} \overset{w}{\rightharpoonup} q_\mathfrak{t}$, *then* $p \xrightarrow{w|\mathfrak{t}'} q$ *and* $\mathfrak{t} \subseteq \mathfrak{t}'$ *for some* $\mathfrak{t}'$.

These lemmas imply the equivalence of TrNFA A and the constructed automaton $A' : L(A) = L(A')$. Thus, pre^* is rational and effectively computable.

7 Related Works

Conditional pushdown systems are introduced for the analysis of programs with runtime inspection [9,12]. The second author of this paper recently applied them to formalize a subset of the HTML5 parser specification [14]. A similar extension of pushdown systems is considered in [7] to formulate abstract garbage collection in the control flow analysis of higher-order programs.

We should clarify the relation between discrete timed pushdown systems of Abdulla et al. [2] and transformable pushdown systems in this paper. Stack symbols of a discrete timed pushdown system are equipped with a natural number representing its *age*, and thus stack is a string over $\Gamma \times \mathbb{N}$. However, as the region construction of timed automata [3], it is sufficient to consider $\mathbb{N}_{\leq m} = \{x \mid 0 \leq x \leq m\} \cup \{\omega\}$ where m is the maximum number appearing in conditions of transitions. Then, a discrete timed pushdown system can be considered as a transformable pushdown system. Abdulla et al. [1] also introduced dense timed pushdown systems and showed that the state reachability problem is decidable through the translation to pushdown systems. The idea of the construction is a combination of the region construction and the construction for TrPDS. However, the construction is very involved and it is not clear whether we can clarify the construction by using TrPDS.

We have extended the saturation procedure to compute pre^* for finite TrPDS by introducing TrNFA. This procedure is closely related to the generalized reachability analysis of pushdown systems with indexed weighted domains [13]. It will be possible to refine the pseudo formal power series semiring in this paper to an indexed semiring and consider the saturation procedure as that for weighted pushdown systems.

In regular model checking [6], transitions of a system are modeled by a length-preserving rational transduction. The verification is conducted by computing the transitive closure of a transduction. From a viewpoint of reachability analysis, our approach and regular model checking are similar but we handle push and pop operations that are not represented by length-preserving rational transductions.

8 Conclusion and Future Works

We have introduced a general framework TrPDS to extend pushdown systems with transitions that modify the whole stack contents with a transducer. The class of finite TrPDS generalizes conditional and transformable pushdown systems, and even a combination of the two systems. A finite TrPDS can be simulated by an ordinary pushdown system, and the saturation procedure for computing pre^* can be extended for finite TrPDS.

We only consider manipulations of stack that can be represented with a length-preserving rational transduction. We believe that the framework of TrPDS can be extended for general rational transductions. However, it will be necessary to revise the definition of the closure based on quotient and the representation of transductions must be taken into account.

Most of our results on TrPDS depend on the finiteness of the closure of a transduction set. Thus, it is natural to ask whether it is decidable to check the closure of a transductions set is finite. As far as we know, this problem has not been investigated yet. We have shown that the following problem is undecidable by using undecidability of *uniformly halting problem* [11]:

> For a given set T of length-preserving rational transductions, decide whether or not the semigroup generated from $(T, \circ)$ is finite.

However, it seems that this result cannot easily be extended for the closure of a set of transductions in this paper.

References

1. Abdulla, P.A., Atig, M.F., Stenman, J.: Dense-timed pushdown automata. In: LICS 2012, pp. 35–44. IEEE Computer Society (2012)
2. Abdulla, P.A., Atig, M.F., Stenman, J.: The minimal cost reachability problem in priced timed pushdown systems. In: Dediu, A.-H., Martín-Vide, C. (eds.) LATA 2012. LNCS, vol. 7183, pp. 58–69. Springer, Heidelberg (2012)
3. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science 126(2), 183–235 (1994)
4. Bouajjani, A., Esparza, J.: Rewriting models of boolean programs. In: Pfenning, F. (ed.) RTA 2006. LNCS, vol. 4098, pp. 136–150. Springer, Heidelberg (2006)
5. 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)
6. Bouajjani, A., Jonsson, B., Nilsson, M., Touili, T.: Regular model checking. In: Emerson, E.A., Sistla, A.P. (eds.) CAV 2000. LNCS, vol. 1855, pp. 403–418. Springer, Heidelberg (2000)
7. Earl, C., Sergey, I., Might, M., Van Horn, D.: Introspective pushdown analysis of higher-order programs. In: ICFP 2012, pp. 177–188. ACM (2012)
8. 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)
9. Esparza, J., Kučera, A., Schwoon, S.: Model checking LTL with regular valuations for pushdown systems. Information and Computation 186(2), 355–376 (2003)
10. Finkel, A., Willems, B., Wolper, P.: A direct symbolic approach to model checking pushdown systems. In: INFINITY 1997. ENTCS, vol. 9, pp. 27–37 (1997)
11. Hughes, C.E., Selkow, S.M.: The finite power property for context-free languages. Theoretical Computer Science 15(1), 111–114 (1981)
12. Li, X., Ogawa, M.: Conditional weighted pushdown systems and applications. In: PEPM 2010, pp. 141–150. ACM (2010)
13. Minamide, Y.: Weighted pushdown systems with indexed weight domains. In: Piterman, N., Smolka, S.A. (eds.) TACAS 2013. LNCS, vol. 7795, pp. 230–244. Springer, Heidelberg (2013)
14. Minamide, Y., Mori, S.: Reachability analysis of the HTML5 parser specification and its application to compatibility testing. In: Giannakopoulou, D., Méry, D. (eds.) FM 2012. LNCS, vol. 7436, pp. 293–307. Springer, Heidelberg (2012)