1 Introduction

2 Preliminaries

Let $\mathbb{N} = \{1, 2, \ldots\}$, $\mathbb{N}_0 = \{0\} \cup \mathbb{N}$ and $[n] = \{1, \cdots, n\}$ for $n \in \mathbb{N}$. For a set A, let $\mathscr{P}(A)$ be the power set of A, A^* and A^ω be the sets of finite and infinite words over A, and we denote $A^\infty = A^* \cup A^\omega$. For a word $\alpha \in A^\infty$ over a set A, let $\alpha(i) \in A$ be the i-th element of α ($i \geq 0$). Let $\langle u, w\rangle = u(0)w(0)u(1)w(1)\cdots \in A^\infty$ for words $u, w \in A^\infty$ and $\langle B, C\rangle = \{\langle u, w\rangle \mid u \in B, w \in C\}$ for sets $B, C \subseteq A^\infty$. By $|\beta|$, we mean the cardinality of β if β is a set and the length of β if β is a finite sequence. We assume ε represents the empty sequence. For $w \in (A \cup \{\varepsilon\})^\omega$, let $ef(w) \in A^\omega$ be an epsilon free sequence of w. That is, $ef(w)(i) = w(i)$ if $w(i) \in A$ and $ef(w)(i) = \varepsilon$ if $w(i) = \varepsilon$ for $i \geq 0$.

In this paper, disjoint sets $\Sigma_{\mathbb{I}}, \Sigma_{\mathbb{O}}$ and Γ denote a (finite) input alphabet, an output alphabet and a stack alphabet, respectively, and $\Sigma = \Sigma_{\mathbb{I}} \cup \Sigma_{\mathbb{O}}$. For a set Γ, let $Com(\Gamma) = \{pop, skip\} \cup \{push(z) \mid z \in \Gamma\}$ be the set of stack commands over Γ. Let $Z : Com(\Gamma) \times \Gamma \to \Gamma^*$ be a function defined as $Z(pop, z) = \varepsilon$, $Z(skip, z) = z$ and $Z(push(z'), z) = z'z$.

2.1 Transition Systems

Definition 1. *A transition system (TS) is* $\mathcal{S} = (S, s_0, A, \to_{\mathcal{S}}, c)$ *where*

- S *is a (finite or infinite) set of states,*
- $s_0 \in S$ *is the initial state,*
- A *is a (finite or infinite) alphabet,*
- $\to_{\mathcal{S}} \subseteq S \times (A \cup \{\varepsilon\}) \times S$ *is a set of transition relation, written as* $s \to^a s'$ *if* $(s, a, s') \in \to_{\mathcal{S}}$ *and*
- $c : S \to [n]$ *is a coloring function where* $n \in \mathbb{N}$.

A run of TS $\mathcal{S} = (S, s_0, A, \to_{\mathcal{S}}, c)$ is a pair $(\rho, w) \in S^\omega \times A^\omega$ that satisfies $\rho(0) = s_0$ and $\rho(i) \to^{w(i)} \rho(i+1)$ for $i \geq 0$. Let $C : S^\omega \to [n]$ be a minimal coloring function such that $C(\rho) = \min\{m \mid$ there exist infinite numbers of $i \geq 0$ such that $c(\rho(i)) = m\}$. We call $\mathcal{S}$ deterministic if $s \to^a s_1$ and $s \to^a s_2$ implies $s_1 = s_2$ for all $s, s_1, s_2 \in S$ and $a \in A \cup \{\varepsilon\}$. We define the language of $\mathcal{S}$ as $L(\mathcal{S}) = \{ef(w) \in A^\omega \mid$ there exists a run (ρ, w) such that $C(\rho)$ is even$\}$.

3 Pushdown Transducers, Automata and Games

3.1 Pushdown Transducers

Definition 2. *A pushdown transducer (PDT) over finite alphabets* $\Sigma_{\mathbb{I}}$, $\Sigma_{\mathbb{O}}$ *and* Γ *is* $\mathcal{T} = (P, p_0, z_0, \Delta)$ *where* P *is a finite set of states,* $p_0 \in P$ *is the initial state,* $z_0 \in \Gamma$ *is the initial stack symbol and* $\Delta : P \times \Sigma_{\mathbb{I}} \times \Gamma \to P \times \Sigma_{\mathbb{O}} \times Com(\Gamma)$ *is a finite set of deterministic transition rules having one of the following forms:*

– $(p, a, z) \to (q, b, pop)$ *(pop rule)*
– $(p, a, z) \to (q, b, skip)$ *(skip rule)*
– $(p, a, z) \to (q, b, push(z))$ *(push rule)*

where $p, q \in P$, $a \in \Sigma_{\mathbb{I}}$, $b \in \Sigma_{\mathbb{O}}$ *and* $z \in \Gamma$.

For a state $p \in P$ and a stack $w \in \Gamma^*$, (p, w) is called a *configuration* or *instantaneous description (abbreviated as ID)* of PDT $\mathcal{T}$. Let $ID_{\mathcal{T}}$ denote the set of all IDs of $\mathcal{T}$. Let $\Rightarrow_{\mathcal{T}} \subseteq ID_{\mathcal{T}} \times \Sigma_{\mathbb{I}} \cdot \Sigma_{\mathbb{O}} \times ID_{\mathcal{T}}$ be the transition relation of $\mathcal{T}$ that satisfies follows: For two IDs $(p, u), (q, u') \in ID_{\mathcal{T}}$ and $ab \in \Sigma_{\mathbb{I}} \cdot \Sigma_{\mathbb{O}}$, $((p, u), ab, (q, u')) \in \Rightarrow_{\mathcal{T}}$, written as $(p, w) \Rightarrow_{\mathcal{T}}^{ab} (q, w')$, if there exist a rule $r = (p, a, z) \to (q, b, com) \in \Delta$ and a sequence $u \in \Gamma^*$ such that $w = zu$ and $w' = Z(com, z)u$. If $\mathcal{T}$ is clear from the context, we abbreviate $\Rightarrow_{\mathcal{T}}^{ab}$ as $\Rightarrow^{ab}$. By definition, any ID $(p, \varepsilon) \in ID_{\mathcal{T}}$ has no successor. That is, there is no transition from an ID with empty stack. We define a run and Language of PDT $\mathcal{T}$ as those of deterministic TS $(ID_{\mathcal{T}}, (q_0, z_0), \Sigma_{\mathbb{I}} \cdot \Sigma_{\mathbb{O}}, \Rightarrow_{\mathcal{T}}, c)$ where $c(s) = 2$ for all $s \in ID_{\mathcal{T}}$. Let **PDT** be the class of PDT.

3.2 Pushdown Automata

Definition 3. *A nondeterministic pushdown automata (NPDA) over finite alphabets* $\Sigma_{\mathbb{I}}$, $\Sigma_{\mathbb{O}}$ *and* Γ *is* $\mathcal{A} = (Q, Q_{\mathbb{I}}, Q_{\mathbb{O}}, q_0, z_0, c, \delta)$ *where* Q, $Q_{\mathbb{I}}, Q_{\mathbb{O}}$ *are finite sets of states that satisfy* $Q = Q_{\mathbb{I}} \cup Q_{\mathbb{O}}$ *and* $Q_{\mathbb{I}} \cap Q_{\mathbb{O}} = \emptyset$, $q_0 \in Q_{\mathbb{I}}$ *is the initial state,* $z_0 \in \Gamma$ *is the initial stack symbol,* $c : Q \to [n]$ *is a coloring function where* $n \in \mathbb{N}$ *is the number of priorities and* $\delta : Q \times \Sigma \times \Gamma \to \mathscr{P}(Q \times Com(\Gamma))$ *is a finite set of transition rules, having one of the following forms:*

– $(q_{\mathbb{X}}, a_{\mathbb{X}}, z) \to (q_{\overline{\mathbb{X}}}, com)$ *(input/output rules)*
– $(q_{\mathbb{X}}, \varepsilon, z) \to (q'_{\mathbb{X}}, com)$ *(ε rules)*

where $(\mathbb{X}, \overline{\mathbb{X}}) \in \{(\mathbb{I}, \mathbb{O}), (\mathbb{O}, \mathbb{I})\}$, $q_{\mathbb{X}}, q'_{\mathbb{X}} \in Q_{\mathbb{X}}, q_{\overline{\mathbb{X}}} \in Q_{\overline{\mathbb{X}}}, a_{\mathbb{X}} \in \Sigma_{\mathbb{X}}$, $z \in \Gamma$ *and* $com \in Com(\Gamma)$.

We define $ID_{\mathcal{A}} = Q \times \Gamma^*$ and a transition relation $\vdash_{\mathcal{A}} \subseteq ID_{\mathcal{A}} \times (\Sigma \cup \{\varepsilon\}) \times ID_{\mathcal{A}}$ as $((q, u), a, (q', u')) \in \vdash_{\mathcal{A}}$ iff there exist a rule $(p, a, z) \to (q, com) \in \delta$ and a sequence $u \in \Gamma^*$ such that $w = zu$ and $w' = Z(com, z)u$. We write $(q, u) \vdash_{\mathcal{A}}^a (q', u')$ iff $((q, u), a, (q', u')) \in \vdash_{\mathcal{A}}$. We write $\vdash_{\mathcal{A}}^a$ as $\vdash^a$ if $\mathcal{A}$ is clear from comtext.

We call $\mathcal{A}$ ε-free if $\mathcal{A}$ has no ε rule. We define a run and language as those of TS $\mathcal{S}_{\mathcal{A}} = (ID_{\mathcal{A}}, (q_0, z_0, \Sigma, \vdash_{\mathcal{A}}, c')$ of $\mathcal{A}$ where $c'((q, u)) = c(q)$ for every $(q, u) \in ID_{\mathcal{A}}$. We call a PDA $\mathcal{A}$ deterministic if $\mathcal{S}_{\mathcal{A}}$ is deterministic, and then we write $\mathcal{A}$ is DPDA. Let **DPDA** and **NPDA** be the class of ε-free DPDA and ε-free NPDA, respectively.

3.3 Pushdown Games

Definition 4. *A Pushdown Games (PDG) of PDA* $\mathcal{A} = (Q, Q_{\mathbb{I}}, Q_{\mathbb{O}}, q_0, z_0, \delta, c)$ *is* $\mathcal{G}_{\mathcal{A}} = (V, V_{\mathbb{I}}, V_{\mathbb{O}}, E, C)$ *where* $V = Q \times \Gamma^*$ *is the set of vertices with* $V_{\mathbb{I}} =$

$Q_{\mathbb{I}} \times \Gamma^*, V_{\mathbb{O}} = Q_{\mathbb{O}} \times \Gamma^*$, $E \subseteq V \times V$ is the set of edges defined as $E = \{(v, v') \mid v \vdash^a v'$ for some $a \in \Sigma\}$ and $C : V \to [n]$ is the coloring function such that $C((q, u)) = c(q)$ for all $(q, u) \in V$.

The game starts with some $(q_0, z_0) \in V_{\mathbb{I}}$. When the current vertex is $v \in V_{\mathbb{I}}$, Player I chooses a successor $v' \in V_{\mathbb{O}}$ of v as the next vertice. When the current vertex is $v \in V_{\mathbb{O}}$, Player II chooses a successor $v' \in V_{\mathbb{I}}$ of v. A finite or infinite sequence $\rho \in V^\infty$ is valid if $\rho(0) = (q_0, z_0)$ and satisfy $(\rho(i-1), \rho(i)) \in E$ for every $i \geq 1$. A play of $\mathcal{G}_{\mathcal{A}}$ is an infinite and valid sequence $\rho \in V^\omega$. A play ρ is winning for Player I iff $state(\rho)$ is even.

By the definition of $\mathcal{G}_{\mathcal{A}}$, every choice of a successor by players can be also expressed as a choice of a pair $(q, com) \in Q \times Com(\Gamma)$. Furthermore, a choice of a successor can be expressed as a choice of $a \in \Sigma$ if $\mathcal{A}$ is deterministic. Thus, every valid sequence $\rho \in V^\infty$ corresponds one-to-one with a sequence $\tau \in (Q \times Com(\Gamma))^\infty$. In detail, for every $i \geq 0$, $\rho(i) = (q, zu), \tau(i) = (q', com)$ and $\rho(i+1) = (q', Z(com, z)u)$ hold for some $q, q' \in Q, z \in \Gamma, u \in \Gamma^*$ and $com \in Com(\Gamma)$. We call τ valid if the corresponding ρ is valid.

Theorem 5. *[Walukiewucz, 2001] If player I has a winning strategy of $\mathcal{G}_{\mathcal{A}}$, we can construct a PDT $\mathcal{T}$ over $Q_{\mathbb{I}} \times Com(\Gamma), Q_{\mathbb{O}} \times Com(\Gamma)$ and an stack alphabet Γ' that gives a winning strategy of $\mathcal{G}_{\mathcal{A}}$. That is, for every $\tau \in L(\mathcal{T})$, the corresponding play $\rho \in V^\infty$ is winning for Player I.*

When $\mathcal{A}$ is deterministic, there is also a one-to-one correspondence between a valid sequence $\rho \in V^\infty$ and a sequence of input and output alphabets $u \in \Sigma^\infty$. In detail, for every $i \geq 0$, $\rho(i) = (q, zu), \rho(i+1) = (q', Z(com, z)u)$ and $(q, u(i), z) \to (q', com) \in \delta$ hold for some $q, q' \in Q, z \in \Gamma, u \in \Gamma^*$ and $com \in Com(\Gamma)$.

By the correspondence, the following lemma holds.

Lemma 6. *A play ρ is winning for Player I iff the corresponding sequence $w \in \Sigma^\omega$ of ρ satisfies $w \in L(\mathcal{A})$.*

In a similar way to Theorem 5, we can obtain the following lemma.

Lemma 7. *If $\mathcal{A}$ is deterministic and player I has a winning strategy of $\mathcal{G}_{\mathcal{A}}$, we can construct a PDT $\mathcal{T}$ over $\Sigma_{\mathbb{I}}, \Sigma_{\mathbb{O}}$ and Γ' that gives a winning strategy of $\mathcal{G}_{\mathcal{A}}$. That is, for every $w \in L(\mathcal{T})$, the corresponding play $\rho \in V^\infty$ is winning for Player I.*

4 Realizability problems for PDA and PDT

For a specification S and an implementation I, we write $I \models S$ if $L(I) \subseteq L(S)$.

Definition 8. *Realizability problem* Real*($\mathcal{S}$, $\mathcal{I}$) for a class of specifications $\mathcal{S}$ and of implementations $\mathcal{I}$: For a specification $S \in \mathcal{S}$, is there an implementation $I \in \mathcal{I}$ such that $I \models S$?*

Theorem 9. Real(**DPDA**, **PDT**) *is decidable.*

Proof. Let $\mathcal{A}$ be a given DPDA. By Lemmas 6 and 7, we can construct a PDT $\mathcal{T}$ such that $\mathcal{T} \models \mathcal{A}$ if player I has a winning strategy for the game $\mathcal{G}_{\mathcal{A}}$. Because there is an algorithm for constructing $\mathcal{T}$ [Walukiewucz, 2001], Real(**DPDA**, **PDT**) is decidable.

Theorem 10. Real(**NPDA**, **PDT**) *is undecidable.*

Proof. For NPDA, we reduce the problem from the universality problem of NPDA, which is undecidable. For a given NPDA $\mathcal{A} = (Q, q_0, z_0, \delta, c)$ over Σ and Γ, we can construct an NPDA $\mathcal{A}' = (Q \cup Q', Q, Q', q_0, z_0, \delta', c')$ over $\Sigma, \Sigma_{\mathbb{O}}$ and Γ that satisfies $L(\mathcal{A}) = \Sigma^\omega$ iff there exists $\mathcal{T}$ such that $\mathcal{T} \models \mathcal{A}$. $\Sigma_{\mathbb{O}}$ is an arbitrary alphabet, $Q' = \{q'_i \mid i \in [n], q_i \in Q\}$ where $Q = \{q_1, \cdots, q_n\}$, $c'(q_i) = c'(q'_i) = c(q_i)$ for all $i \in [n]$ and δ' satisfies that $(q_i, a, z) \to (q_j, com) \in \delta$ iff $(q_i, a, z) \to (q'_j, com) \in \delta'$, and $(q'_j, b, z) \to (q_j, skip) \in \delta'$ for all $b \in \Sigma_{\mathbb{O}}$. By the construction of $\mathcal{A}'$, $L(\mathcal{A}') = \langle L(\mathcal{A}), \Sigma_{\mathbb{O}}^\omega \rangle$ holds. If $L(\mathcal{A}) = \Sigma^\omega$, then $L(\mathcal{A}') = \langle \Sigma^\omega, \Sigma_{\mathbb{O}}^\omega \rangle$ and thus $\mathcal{T} \models \mathcal{A}$ holds for every $\mathcal{T}$. If $L(\mathcal{A}) \neq \Sigma^\omega$, there exists a word $w \in \Sigma^\omega$ such that $w \notin L(\mathcal{A})$. Every language of PDT contains a word $\langle u, v \rangle$ for every $u \in \Sigma^\omega$ and some $v \in \Sigma_{\mathbb{O}}^\omega$, but $\langle w, v \rangle \notin L(\mathcal{A}')$ for any $v \in \Sigma_{\mathbb{O}}^\omega$. Hence, $\mathcal{T} \not\models \mathcal{A}'$ holds for any PDT $\mathcal{T}$. In conclusion, this reduction holds and the realizability problem for PDT and NPDA is undecidable.

5 Register Pushdown Transducers and Register Pushdown Automata

5.1 Data words and registers

We assume a countable set D of *data values*. For finite alphabets $\Sigma_{\mathbb{I}}, \Sigma_{\mathbb{O}}$ and a countable set D, an infinite sequence $(a_1^{\mathbb{I}}, d_1)(a^{\mathbb{O}}, d'_1) \cdots \in ((\Sigma_{\mathbb{I}} \times D) \cdot (\Sigma_{\mathbb{O}} \times D))^\omega$ is called a *data word*. We write $DW(\Sigma_{\mathbb{I}}, \Sigma_{\mathbb{O}}, D) = ((\Sigma_{\mathbb{I}} \times D) \cdot (\Sigma_{\mathbb{O}} \times D))^\omega$.

For $k \in \mathbb{N}_0$, a mapping $\theta : [k] \to D$ is called an *assignment* (of data values to k registers). Let Θ_k denote the collection of assignments to k registers. We specify $\bot \in D$ as the initial data value and $\theta_\bot \in \Theta_k$ be the initial assignment such that $\theta_\bot(i) = \bot$ for all $i \in [k]$.

We denote $Tst_k = \mathscr{P}([k] \cup \{top\})$ and $Asgn_k = \mathscr{P}([k])$ where $top \notin \mathbb{N}$ is the unique symbol that represents a stack top value. Tst_k is the set of guard conditions. For $tst \in Tst_k$, $\theta \in \Theta_k$ and $d, e \in D$, we denote $\theta, d, e \models tst$ if $\theta(i) = d \Leftrightarrow i \in tst$ and $e = d \Leftrightarrow top \in tst$ hold. (In definitions of register pushdown transducer (automaton) in the next section, the data values d and e represent an input data value and a stack top data value, respectively.) $Asgn_k$ is the set of assignment conditions. For $asgn \in Asgn_k$, $\theta, \theta' \in \Theta_k$ and $d \in D$, let $\theta[asgn \leftarrow d]$ be the assignment θ' such that $\theta'(i) = d$ for $i \in asgn$ and $\theta'(i) = \theta(i)$ for $i \notin asgn$. Let $Z_D : Com([k]) \times \Theta_k \times D \to D^*$ be a function defined as $Z_D(pop, \theta, d) = \varepsilon$, $Z_D(skip, \theta, d) = d$ and $Z_D(push(j), \theta, d) = \theta(j)d$.

5.2 Register pushdown transducers

Definition 11. *A k-register pushdown transducer (k-RPDT) over finite alphabets $\Sigma_{\mathbb{I}}, \Sigma_{\mathbb{O}}$ and an infinite set D of data values is $\mathcal{T} = (P, p_0, \Delta)$ where P is a finite set of states, $p_0 \in P$ is the initial state, $\Delta : P \times \Sigma_{\mathbb{I}} \times Tst_k \to P \times \Sigma_{\mathbb{O}} \times Asgn_k \times [k] \times Com([k])$ is a finite set of deterministic transition rules.*

D is used as a stack alphabet. Let $ID_{\mathcal{T}} = P \times \Theta_k \times D^*$ and $\Rightarrow_{\mathcal{T}} \subseteq ID_{\mathcal{T}} \times ((\Sigma_{\mathbb{I}} \times D) \cdot (\Sigma_{\mathbb{O}} \times D)) \times ID_{\mathcal{T}}$ be a transition relation of $\mathcal{T}$ such that $((p, \theta, u), (a, d^{\mathbb{I}})(b, d^{\mathbb{O}}), (q, \theta', u')) \in \Rightarrow_{\mathcal{T}}$ iff there exist a rule $(p, a, tst) \to (q, b, asgn, j, com) \in \Delta$ a data value $e \in D$, and a sequence of data values $w \in D^*$ where $u = ew$ that satisfy the follows: $d^{\mathbb{I}}, e, \theta \models tst$, $\theta' = \theta[asgn \leftarrow d^{\mathbb{I}}]$, $\theta'(j) = d^{\mathbb{O}}$ and $u' = Z_D(com, \theta', e)w$, and then we write $(p, \theta, u) \Rightarrow_{\mathcal{T}}^{(a,d^{\mathbb{I}})(b,d^{\mathbb{O}})} (q, \theta', u')$. If $\mathcal{T}$ is clear from the context, we abbreviate $\Rightarrow_{\mathcal{T}}^{(a,d^{\mathbb{I}})(b,d^{\mathbb{O}})}$ as $\Rightarrow^{(a,d^{\mathbb{I}})(b,d^{\mathbb{O}})}$.

The run and languages of $\mathcal{T}$ is those of TS $(ID_{\mathcal{T}}, (q_0, \theta_{\bot}, \bot), (\Sigma_{\mathbb{I}} \times D) \cdot (\Sigma_{\mathbb{O}} \times D), \Rightarrow_{\mathcal{T}}, c)$ where $c(s) = 2$ for all $s \in ID_{\mathcal{T}}$. Let $\mathbf{RPDT}[k]$ be the class of k-RPDT and $\mathbf{RPDT} = \bigcup_{k \in \mathbb{N}_0} \mathbf{RPDT}[k]$.

5.3 Register pushdown automata

Definition 12. *A nondeterministic k-register pushdown automaton (k-NRPDA) over $\Sigma_{\mathbb{I}}, \Sigma_{\mathbb{O}}$ and D is $\mathcal{A} = (Q, Q_{\mathbb{I}}, Q_{\mathbb{O}}, q_0, \delta, c)$, where*

- *Q is a finite set of states,*
- *$Q_{\mathbb{I}} \cup Q_{\mathbb{O}} = Q, Q_{\mathbb{I}} \cap Q_{\mathbb{O}} = \emptyset$,*
- *$q_0 \in Q$ is the initial state, and*
- *$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times Tst_k \to \mathscr{P}(Q \times Asgn_k \times Com([k]))$ is a transition function having one of the forms:*
  - *$(q_{\mathbb{X}}, a_{\mathbb{X}}, tst) \to (q_{\overline{\mathbb{X}}}, asgn, com)$ (input rule)*
  - *$(q_{\mathbb{X}}, \varepsilon, tst) \to (q'_{\mathbb{X}}, asgn, com)$ (ε rule)*

  where $(\mathbb{X}, \overline{\mathbb{X}}) \in \{(\mathbb{I}, \mathbb{O}), (\mathbb{O}, \mathbb{I})\}$, $q_{\mathbb{X}}, q'_{\mathbb{X}} \in Q_{\mathbb{X}}, q_{\overline{\mathbb{X}}} \in Q_{\overline{\mathbb{X}}}, a_{\mathbb{X}} \in \Sigma_{\mathbb{X}}$, $tst \in Tst_k$, $asgn \in Asgn_k$ and $com \in Com([k])$.
- *$c : Q \to [n]$ where $n \in \mathbb{N}$ is the number of priorities.*

Let $ID_{\mathcal{A}} = Q \times \Theta_k \times D^*$. We define a set of transition relation $\vdash_{\mathcal{A}} \subseteq ID_{\mathcal{A}} \times ((\Sigma \cup \{\varepsilon\}) \times D) \times ID_{\mathcal{A}}$ as satisfying $((q, \theta, u), (a, d), (q', \theta', u')) \in \vdash_{\mathcal{A}}$, written as $(q, \theta, u) \vdash^{(a,d)} (q', \theta', u')$, iff there exist a rule $(p, a, tst) \to (q, asgn, com) \in \delta$ a data value $e \in D$, and a sequence of data values $w \in D^*$ where $u = ew$ that satisfy the follows: $d, e, \theta \models tst$, $\theta' = \theta[asgn \leftarrow d]$ and $u' = Z_D(com, \theta', e)w$. We write $\vdash_{\mathcal{A}}^{(a,d)}$ as $\vdash^{(a,d)}$ if $\mathcal{A}$ is clear from comtext. The run and language of k-DRPDA $\mathcal{A}$ is those of TS $\mathcal{S}_{\mathcal{A}} = (ID_{\mathcal{A}}, (q_0, \theta_{\bot}, \bot), (\Sigma \cup \{\varepsilon\}) \times D, \Rightarrow_{\mathcal{A}}, c')$ where $c'((q, \theta, u)) = c(q)$ for all $(q, \theta, u) \in ID_{\mathcal{A}}$. We call $\mathcal{A}$ deterministic, or k-DRPDA, if $\mathcal{S}_{\mathcal{A}}$ is deterministic.

5.4 Classes of RPDA

An ε-free k-RPDA is an RPDA not having any ε rules. Let **DRPDA** and **NRPDA** be the class of ε-free k-DRPDA and k-NRPDA for all $k \in \mathbb{N}_0$, respectively. Let $Com_v = \{pop, skip, push\}$ and $v : Com([k]) \to Com$ be a function such that $v(push(j)) = push$ for $j \in [k]$ and $v(com) = com$ otherwise. An visible k-RPDA is input and output alphabets are $\Sigma_{\mathbb{I}} \times Com_v$ and $\Sigma_{\mathbb{O}} \times Com_v$, respectively, and every rule has one of the form $(q, (a, v(com))) \to (q', asgn, com)$. Let **DRPDAv** be the class of visible ε-free k-DRPDA for all $k \in \mathbb{N}_0$, respectively.

6 Realizability problems for RPDA and RPDT

6.1 Finite actions

For $k \in \mathbb{N}_0$, we define the set of visible finite input actions as $A_k^{\mathbb{I}} = \Sigma_{\mathbb{I}} \times \{skip\} \times Tst_k$ and the set of visible output actions as $A_k^{\mathbb{O}} = \{(\sigma_o, v(com), asgn, j, com) \in \Sigma_{\mathbb{O}} \times Com_v \times Asgn_k \times [k] \times Com([k])\}$ for k-RPDT. A sequence $w = ((a_1^{\mathbb{I}}, skip), d_1^{\mathbb{I}})((a_1^{\mathbb{O}}, v(com_1)), d_1^{\mathbb{O}}) \cdots \in DW(\Sigma_{\mathbb{I}}, \Sigma_{\mathbb{O}}, D)$ is compatible with a sequence $\overline{a} = (a_1^{\mathbb{I}}, skip, tst_1)(a_1^{\mathbb{O}}, v(com_1), asgn_1, j_1, com_1) \cdots \in (A_k^{\mathbb{I}} A_k^{\mathbb{O}})^\omega$ if there exists a run (ρ, w) of k-RPDT satisfying follows: For all $i \geq 1$, let $\rho(i-1) = (q, \theta, eu)$ and $\rho(i) = (q', \theta', u'u)$ for some $e \in D, u \in D^*$ and $u' \in D^*$. Then $\theta, d_i^{\mathbb{I}}, e \models tst_i, \theta' = \theta[asgn_i \leftarrow d_i^{\mathbb{I}}], \theta'(j_1) = d_i^{\mathbb{O}}$ and $u' = Z_D(com, \theta', e)$ hold. Let $Comp(\overline{a}) = \{w \in DW(\Sigma_{\mathbb{I}}, \Sigma_{\mathbb{O}}, D) \mid w \text{ is compatible with } \overline{a}\ \}$. For specification $S \subseteq DW(\Sigma_{\mathbb{I}}, \Sigma_{\mathbb{O}}, D)$, we define $W_{S,k} = \{\overline{a} \mid Comp(\overline{a}) \subseteq S\}$.

Theorem 13. *For a specification $S \subseteq DW(\Sigma_{\mathbb{I}}, \Sigma_{\mathbb{O}}, D)$, the following statements are equivalent.*

- *There exists a k-RPDT $\mathcal{T}$ such that $L(\mathcal{T}) \subseteq S$.*
- *There exists a PDT $\mathcal{T}'$ such that $L(\mathcal{T}') \subseteq W_{S,k}$.*

6.2 Decidability and undecidability of realizability problems

Lemma 14. *$L_k = \{w \otimes \overline{a} \mid w \in Comp(\overline{a})\}$ is definable as a language of $(k+2)$-DRPDA.*

Proof. Let $(k+2)$-DRPDA $\mathcal{A}_k = (\{p, q\} \cup (Asgn_k \times [k] \times Com([k])) \cup [k], \{p\}, \{q\} \cup (Asgn_k \times [k] \times Com([k])) \cup [k], p, \delta_k, c_k)$ over $A_k^{\mathbb{I}}, A_k^{\mathbb{O}}$ and D where $c_k(s) = 2$ for all state s and δ_k consists of rules of the form

$$(p, (a_{\mathbb{I}}, skip, tst), tst) \to (q, \{k+1\}, skip) \tag{1}$$

$$(q, (a_{\mathbb{O}}, v(com), asgn, j, com), tst') \to ((asgn, j, com), \{k+2\}, skip) \tag{2}$$

$$((asgn, j, com), \varepsilon, \{k+1\} \cup tst') \to (j, asgn, com) \tag{3}$$

$$(j, \varepsilon, \{j, k+2\} \cup tst') \to (p, \emptyset, skip) \tag{4}$$

for all $(a_{\mathbb{I}}, tst) \in A_k^{\mathbb{I}}$, $(a_{\mathbb{O}}, asgn, j, com) \in A_k^{\mathbb{O}}$ and $tst' \in Tst_{k+2}$. Then, $L(\mathcal{A}_k) = L_k$ holds.

Lemma 15. *For specification $\mathcal{S}$ definable by some visible ε-free k'-DRPDA. $L_{k,\overline{S}} = \{w \otimes \overline{a} \mid w \in Comp(\overline{a}) \cap \overline{S}\}$ is definable as a language of visible $(k+k'+4)$-DRPDA.*

Proof. Let $L_{\overline{S}} = \{w \otimes \overline{a} \mid w \in \overline{S}\}$, $\mathcal{A}_{\overline{S}}$ be a visible ε-free k'-DRPDA such that $L(\mathcal{A}_{\overline{S}}) = L_{\overline{S}}$ and $\mathcal{A}_k$ be a $(k+2)$-DRPDA defined in Lemma 14. Because $L_{k,\overline{S}} = L_k \cap L_{\overline{S}}$ and both L_k and $L_{\overline{S}}$ are visible DRPDA, it is enough to show we can construct visible $(k+k'+4)$-DRPDA $\mathcal{A}$ such that $L(\mathcal{A}) = L(\mathcal{A}_{\overline{S}}) \cap L(\mathcal{A}_k)$.

For simplicity, we rewrite $\mathcal{A}_k$ as k_1-DRPDA $\mathcal{A}_1 = (Q_1, Q_1^{\mathbb{I}}, Q_1^{\mathbb{O}}, q_1^0, \delta_1, c_1)$ and $\mathcal{A}_{\overline{S}}$ as k_2-DRPDA $\mathcal{A}_2 = (Q_2, Q_2^{\mathbb{I}}, Q_2^{\mathbb{O}}, q_2^0, \delta_2, c_2)$, but they satisfy that $c_1(q)$ is even for all $q \in Q_1$ and every rules in δ_1 forms triple sequencial rules

$$(q_1, (a, v(com_1)), tst_1) \to (q_2, asgn_1, skip) \tag{2'}$$
$$(q_2, \varepsilon, tst_2) \to (q_3, asgn_2, com_1) \tag{3'}$$
$$(q_3, \varepsilon, tst_3) \to (q_4, asgn_3, skip) \quad . \tag{4'}$$

Note that $(2'), (3')$ and $(4')$ correspond to $(2), (3)$ and (4), respectively, and (1) can be devided in three rules of the form $(2'), (3')$ and $(4')$.

We construct (k_1+k_2+2)-DRPDA $\mathcal{A} = (Q_1 \times Q_2 \times [5], Q_1^{\mathbb{I}} \times Q_2^{\mathbb{I}} \times [5], Q_1^{\mathbb{O}} \times Q_2^{\mathbb{O}} \times [5], (q_0^1, q_0^2, 1), \delta, c)$ where $c((q_1, q_2, i)) = c_2(q_2)$ for all $(q_1, q_2, i) \in Q$. For all rules

- $(q_1, (a, v(com_1)), tst_1) \to (q_2, asgn_1, skip)$,
- $(q_2, \varepsilon, tst_2) \to (q_3, asgn_2, com_1)$,
- $(q_3, \varepsilon, tst_3) \to (q_4, asgn_3, skip) \in \delta_1$ and
- $(q, (a, v(com)), tst) \to (q', asgn, com) \in \delta_2$

$(v(com_1) = v(com))$ for $a \in \Sigma$, let $tst^{+k_1} = \{i + k_1 \mid i \in tst\} \cup \{top \mid top \in tst \setminus [k_1]\}$, $asgn^{+k_1} = \{i + k_1 \mid i \in asgn\}$ and $com^{+k_1} = push(j + k_1)$ if $com = push(j)$ and $com^{+k_1} = com$ otherwise, then δ consists of the rules

- $((q_1, q, 1), \varepsilon, tst' \cup \{top\}) \to ((q_1, q, 2), \{k_1 + k_2 + 1\}, pop)$
- $((q_1, q, 2), \varepsilon, tst' \cup \{top\}) \to ((q_1, q, 3), \{k_1 + k_2 + 2\}, push(k_1 + k_2 + 1))$
- $((q_1, q, 3), (a, v(com_1)), (tst_1 \cup tst^{+k_1}) \setminus top \cup \{k_1 + k_2 + t \mid t = 1$ if $top \in tst_1$ and $t = 2$ if $top \in tst$ $\}) \to ((q_2, q', 4), asgn_1 \cup asgn^{+k_1}, com^{+k_1})$
- $((q_2, q', 4), \varepsilon, tst_2 \cup tst') \to ((q_3, q', 5), asgn_2, com_1)$
- $((q_3, q', 5), \varepsilon, tst_3 \cup tst') \to ((q_4, q', 0), asgn_3, skip)$

for all $tst' \in Tst_{k_1+k_2+2}$. Then, $L(\mathcal{A}) = L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$.

Lemma 16. $W_{S,k} = \overline{Lab(L_{\overline{S},k})}$.

Proof. For every $\overline{a} \in (A_k^{\mathbb{I}} A_k^{\mathbb{O}})^\omega$, $\overline{a} \notin W_{S,k} \Leftrightarrow Comp(\overline{a}) \not\subseteq S \Leftrightarrow \exists w.w \in Comp(\overline{a}) \cap \overline{S} \Leftrightarrow \exists w.w \otimes \overline{a} \in L_{\overline{S},k} \Leftrightarrow \overline{a} \in Lab(L_{\overline{S},k})$. Thus, $W_{S,k} = \overline{Lab(L_{\overline{S},k})}$ holds.

Theorem 17. *For all $k \geq 0$, $\textsc{Real}(\mathbf{DRPDAv}, \mathbf{RPDT}[k])$ is decidable.*

Proof. By Lemma 15, $L_{\overline{S},k}$ is definable by some visible DRPDA. Because every language recognized by some visible DRPDA can be converted to the language of visible DPDA by taking a projection on its label, $W_{S,k}$ is definable by some visible DPDA by Lemma 16. By Theorem 13, we can check Real(**DPDA**, **PDT**) for $W_{S,k}$, which is shown to be decidable in Theorem 9, instead of checking Real(**DRPDAv**, **RPDT**$[k]$).

Theorem 18. *For all $k \geq 0$,* Real(**NRPDA***,* **RPDT**$[k]$) *is undecidable.*

Proof. We can easily reduce the problem from Real(**NPDA**, **PDT**), whose undecidability is proved in Theorem 10.

7 Conclusion

References