Nested Timed Automata with Frozen Clocks

Guoqiang Li[1], Mizuhito Ogawa[2], and Shoji Yuen[3]

[1] BASICS, School of Software, Shanghai Jiao Tong University, China
li.g@sjtu.edu.cn
[2] Japan Advanced Institute of Science and Technology, Japan
mizuhito@jaist.ac.jp
[3] Graduate School of Information Science, Nagoya University, Japan
yuen@is.nagoya-u.ac.jp

Abstract. A nested timed automaton (NeTA) is a pushdown system whose control locations and stack alphabet are *timed automata (TAs)*. A control location describes a working TA, and the stack presents a pile of interrupted TAs. In NeTAs, all local clocks of TAs proceed uniformly also in the stack. This paper extends NeTAs with frozen local clocks (NeTA-Fs). All clocks of a TA in the stack can be either frozen or proceeding when it is pushed. A NeTA-F also allows global clocks adding to local clocks in the working TA, which can be referred and/or updated from the working TA. We investigate the reachability of NeTA-Fs showing that (1) the reachability with a single global clock is *decidable*, and (2) the reachability with multiple global clocks is *undecidable*.

1 Introduction

Recently, modeling and analyzing complex real-time systems with recursive context switches have attracted attention. Difficulty on decidability of crucial properties, e.g. safety, comes from two dimensions of infinity, an unboundedly large stack and various types of clocks that record dense time.

Timed automata (TAs) [1] are finite automata with a finite set of *clocks*, of which the constant slope is always 1. A special type of a clock is a stopwatch, which has either 1 or 0 as the constant slope. A *stopwatch automaton* is a TA with stopwatches, and surprisingly its reachability becomes undecidable [5].

For a component-based recursive timed system, clocks are naturally classified into *global clocks*, which can be updated and observed by all contexts, and *local clocks*, which belong to the context of a component and will be stored in the stack when the component is interrupted. Similar to stopwatches, we introduce a special type of local clocks, named *frozen clocks*, whose values are not updated while their context is preempted and restart update when resumed. Other local clocks are *proceeding*. The reachability of a recursive timed systems are investigated in various models, such as *recursive timed automata (RTAs)* [2], *timed recursive state machines (TRSMs)* [3], and *nested timed automata (NeTAs)* [4]. Recently, RTAs are extended to *recursive hybrid automata (RHA)* [7].

Both RTAs [2] and TRSMs [3] adopt timed state machines as a formalization, which is regarded as a TA with explicit entry and exit states. In both models, each timed state machine (TSM) shares the same set of clocks. To guarantee the decidability of the reachability, RTAs restrict all clocks to be either call-by-value or call-by-reference, in our terminology frozen or global clocks respectively. TRSMs are restricted to be either local or initialized. Local TRSMs restore the values of all clocks when a pop occurs. Initialized TRSMs reset all clocks to zero when a push occurs. The clocks in local-TRSMs are regarded as frozen clocks, while those in initialized-TRSMs are special cases of global clocks.

Similar to stopwatches, frozen clocks significantly affect the decidability of the reachability, observed by encoding counters with the N-wrapping technique (Fig. 1 in Section 4.2). The recursive timed systems above either prohibit to pass values between clocks and stopwatches, or have no stopwatches. Thus, they avoid the wrapping technique and the reachability remains decidable. Note that the wrapping technique is avoided if a TA has a single stopwatch (without other clocks). *Interrupt timed automata* [6] push such a stopwatch automaton into the stack, and the single stopwatch restriction preserves the decidable reachability.

This paper investigates the decidability of the reachability of *NeTAs with frozen clocks* (NeTA-Fs), which have all three types of clocks. All local clocks of a TA in a NeTA-F are either frozen or proceeding when the TA is pushed to the stack. Moreover, global clocks may exchange values with local clocks in the working TA. We show that (1) the reachability with a single global clock is *decidable*, and (2) the reachability with multiple global clocks is *undecidable*.

NeTA-Fs naturally express interrupt behavior with time as follows. At the moment of interrupt, the current working component is pushed to the stack (its local clocks are either proceeding or frozen), and a handler component starts with the initial setting. When the handler component is finished, the suspended component is popped from the stack to be resumed. Global clocks together with local clocks in the working TA work as proceeding clocks to specify time constraints as well as channels by value passing among components.

The decidability for a NeTA-F with a single global clock is shown by two steps encoding: (1) to an extension of a *dense timed pushdown automaton (DTPDA)* [8, 9] with frozen ages (DTPDA-F), and (2) its digitization a *snapshot pushdown systems* (snapshot PDS), which is a *well-structured pushdown system* [10, 11] with a *well-formed constraint* [9]. Both encoding steps preserve the reachability. The undecidability of the reachability follows from simulating a Minsky machine by a NeTA-F with two global clocks, applying the N-wrapping technique [17].

The rest of the paper is organized as follows. Section 2 recalls TAs and DTPDAs, and then introduces DTPDA-Fs. Section 3 proves the decidability of the reachability of DTPDA-F with a single global clock. Section 4 presents NeTA-F, and proves its decidability and undecidability results depending on the number of global clocks. Section 5 concludes the paper.

2 Dense Timed Pushdown Automata with Frozen Ages

For finite words $w = aw'$, we denote $a = head(w)$ and $w' = tail(w)$. The concatenation of two words w, v is denoted by $w.v$, and ϵ is the empty word.

Let $\mathbb{R}^{\geq 0}$ and $\mathbb{N}$ be the sets of non-negative real and natural numbers, respectively. Let $\mathbb{N}_\omega := \mathbb{N} \cup \{\omega\}$, where ω is the least limit ordinal. $\mathcal{I}$ denotes the set of *intervals*, which are (a, b), $[a, b]$, $[a, b)$ or $(a, b]$ for $a \in \mathbb{N}$ and $b \in \mathbb{N}_\omega$.

Let $X = \{x_1, \ldots, x_n\}$ be a finite set of *clocks*. A *clock valuation* $\nu : X \to \mathbb{R}^{\geq 0}$, assigns a value to each clock $x \in X$. $\nu_\mathbf{0}$ represents all clocks in X assigned to 0. Given a clock valuation ν and a time $t \in \mathbb{R}^{\geq 0}$, $(\nu + t)(x) = \nu(x) + t$, for $x \in X$. A clock assignment function $\nu[y \leftarrow b]$ is defined by $\nu[y \leftarrow b](x) = b$ if $x = y$, and $\nu(x)$ otherwise. $\mathcal{V}al(X)$ is used to denote the set of clock valuation of X.

2.1 Dense timed pushdown automata

Dense timed pushdown automata [8] extend timed pushdown automata with time update in the stack. Each symbol in the stack is equipped with a local clock named an *age*, and all ages in the stack proceed uniformly. An age in each context is assigned to the value of a clock when a push action occurs. A pop action pops the top symbol to assign the value of its age to a specified clock.

Note that, by deleting **push** and **pop** actions (as well as Γ) from a DTPDA, we obtain a timed automaton (TA) [1, 12].

Definition 1 (Dense Timed Pushdown Automata). *A dense timed pushdown automaton is a tuple* $\mathcal{A} = \langle Q, q_0, \Gamma, X, \Delta \rangle \in \mathscr{A}$*, where*

- Q *is a finite set of control states with the initial state* $q_0 \in Q$*,*
- Γ *is finite stack alphabet,*
- X *is a finite set of clocks, and*
- $\Delta \subseteq Q \times Actions \times Q$ *is a finite set of actions.*

A (discrete) transition $\delta \in \Delta$ *is a sequence of actions* $(q_1, \varphi_1, q_2), \cdots, (q_i, \varphi_i, q_{i+1})$ *written as* $q_1 \xrightarrow{\varphi_1; \cdots; \varphi_i} q_{i+1}$*, in which* φ_j *(for* $1 \leq j \leq i$*) is one of the followings,*

- **Local** ϵ*, an empty operation,*
- **Test** $x \in I?$*, where* $x \in X$ *is a clock and* $I \in \mathcal{I}$ *is an interval,*
- **Assign** $x \leftarrow I$ *where* $x \in X$ *and* $I \in \mathcal{I}$*,*
- **Value passing** $x \leftarrow x'$ *where* $x, x' \in X$*.*
- **Push** $push(\gamma, x)$*, where* $\gamma \in \Gamma$ *is a stack symbol and* $x \in X$*, and*
- **Pop** $pop(\gamma, x)$*, where* $\gamma \in \Gamma$ *is a stack symbol and* $x \in X$*.*

A transition as a sequence of actions $q_1 \xrightarrow{\varphi_1; \cdots; \varphi_i} q_{i+1}$ prohibits interleaving time progress. This can be encoded with an extra clock by resetting it to 0 and checking it still 0 after transitions, and introducing fresh control states.

Given a DTPDA $\mathcal{A} \in \mathscr{A}$, we use $Q(\mathcal{A})$, $q_0(\mathcal{A})$, $X(\mathcal{A})$ and $\Delta(\mathcal{A})$ to represent the set of control states, the initial state, the set of clocks and the set of transitions, respectively. We will use similar notations throughout the paper.

Definition 2 (Semantics of DTPDA). *For a dense timed pushdown automaton* $\langle Q, q_0, \Gamma, X, \Delta \rangle$*, a configuration is a triplet* (q, w, ν) *with* $q \in Q$*,* $w \in (\Gamma \times \mathbb{R}^{\geq 0})^*$*, and a clock valuation* ν *on* X*. Time passage of the stack* $w + t = (\gamma_1, t_1 + t). \cdots .(\gamma_n, t_n + t)$ *for* $w = (\gamma_1, t_1). \cdots .(\gamma_n, t_n)$*.*

The transition relation of a DTPDA consists of time progress and a discrete transition which is defined by that of actions below.

– Time progress: $(q, w, \nu) \xrightarrow{t}_{\mathscr{A}} (q, w + t, \nu + t)$, *where* $t \in \mathbb{R}^{\geq 0}$.
– Discrete transition: $(q_1, w_1, \nu_1) \xrightarrow{\varphi}_{\mathscr{A}} (q_2, w_2, \nu_2)$, *if* $q_1 \xrightarrow{\varphi} q_2$, *and one of the following holds,*
  - **Local** $\varphi = \epsilon$, *then* $w_1 = w_2$, *and* $\nu_1 = \nu_2$.
  - **Test** $\varphi = x \in I?$, *then* $w_1 = w_2$, $\nu_1 = \nu_2$ *and* $\nu_1(x) \in I$ *holds.*
  - **Assign** $\varphi = x \leftarrow I$, *then* $w_1 = w_2$, $\nu_2 = \nu_1[x \leftarrow r]$ *where* $r \in I$.
  - **Value passing** $\varphi = x \leftarrow x'$, *then* $w_1 = w_2$, $\nu_2 = \nu_1[x \leftarrow \nu_1(x')]$.
  - **Push** $\varphi = push(\gamma, x)$, *then* $\nu_1 = \nu_2$, $w_2 = (\gamma, \nu_1(x)).w_1$.
  - **Pop** $\varphi = pop(\gamma, x)$, *then* $\nu_2 = \nu_1[x \leftarrow t]$, $w_1 = (\gamma, t).w_2$.

The initial configuration $\varrho_0 = (q_0, \epsilon, \nu_{\mathbf{0}})$.

Remark 1. For simplicity of the later proofs, the definition of DTPDAs is slightly modified from the original [8]. **Value-passing** is introduced; instead $push(\gamma, I)$ and $pop(\gamma, I)$ are dropped, since they are described by $(x \leftarrow I; push(\gamma, x))$ and $(pop(\gamma, x); x \in I?)$, respectively.

2.2 DTPDAs with Frozen Ages

A DTPDA with frozen ages (DTPDA-F) is different from Definition 1 at:

– clocks are partitioned into the set X of local clocks (of the fixed number k) and the set C of global clocks,
– a tuple of ages (for simplicity, we fix the length of a tuple to be k) is pushed on the stack and/or popped from the stack, and
– each tuple of ages is either *proceeding* (as in Definition 1) or *frozen*. After pushing the tuple, all local clocks are reset to zero.

Definition 3 (DTPDAs with Frozen Ages). *A DTPDA with frozen ages (DTPDA-F) is a tuple* $\mathcal{D} = \langle S, s_0, \Gamma, X, C, \Delta \rangle \in \mathscr{D}$, *where*

– S *is a finite set of states with the initial state* $s_0 \in S$,
– Γ *is finite stack alphabet,*
– X *is a finite set of local clocks (with* $|X| = k$*),*
– C *is a finite set of global clocks, and*
– $\Delta \subseteq S \times Action^F \times S$ *is a finite set of actions.*

A (discrete) transition $\delta \in \Delta$ *is a sequence of actions* $(s_1, \varphi_1, s_2), \cdots, (s_i, \varphi_i, s_{i+1})$ *written as* $s_1 \xrightarrow{\varphi_1; \cdots; \varphi_i} s_{i+1}$, *in which* φ_j *(for* $1 \leq j \leq i$*) is one of the followings,*

– **Local** ϵ, *an empty operation,*
– **Test** $x \in I?$, *where* $x \in X \cup C$ *is a clock and* $I \in \mathcal{I}$ *is an interval,*
– **Assign** $x \leftarrow I$ *where* $x \in X \cup C$ *and* $I \in \mathcal{I}$,
– **Value passing** $x \leftarrow x'$ *where* $x, x' \in X \cup C$.
– **Push** $push(\gamma)$, *where* $\gamma \in \Gamma$,
– **Freeze-Push (F-Push)** $fpush(\gamma)$, *where* $\gamma \in \Gamma$, *and*
– **Pop** $pop(\gamma)$, *where* $\gamma \in \Gamma$.

Definition 4 (Semantics of DTPDA-F). *For a DTPDA-F $\langle S, s_0, \Gamma, X, C, \Delta\rangle$, a configuration is a triplet (s, w, ν) with $s \in S$, $w \in (\Gamma \times (\mathbb{R}^{\geq 0})^k \times \{0,1\})^*$, and a clock valuation ν on $X \cup C$. For $w = (\gamma_1, \bar{t}_1, flag_1). \cdots .(\gamma_n, \bar{t}_n, flag_n)$, t-time passage on the stack, written as $w + t$, is $(\gamma_1, progress(\bar{t}_1, t, flag_1), flag_1). \cdots .(\gamma_n, progress(\bar{t}_n, t, flag_n), flag_n)$ where*

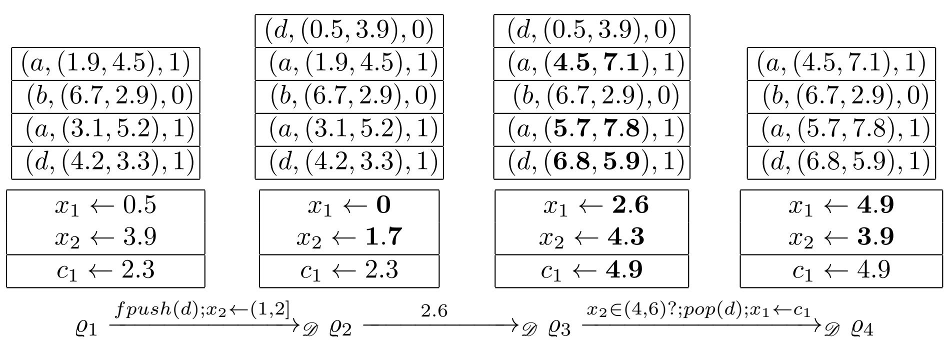
$$progress(\bar{t}, t, flag) = \begin{cases} (t_1 + t, \cdots, t_k + t) & \text{if } flag = 1 \text{ and } \bar{t} = (t_1, \cdots, t_k) \\ \bar{t} & \text{if } flag = 0 \end{cases}$$

The transition relation consists of time progress and a discrete transition.

- Time progress*: $(s, w, \nu) \xrightarrow{t}_{\mathscr{D}} (s, w + t, \nu + t)$, where $t \in \mathbb{R}^{\geq 0}$.*
- Discrete transition*: $(s_1, w_1, \nu_1) \xrightarrow{\varphi}_{\mathscr{D}} (s_2, w_2, \nu_2)$, if $s_1 \xrightarrow{\varphi} s_2$, and one of the following holds,*
  - **Local** *$\varphi = \epsilon$, then $w_1 = w_2$, and $\nu_1 = \nu_2$.*
  - **Test** *$\varphi = x \in I?$, then $w_1 = w_2$, $\nu_1 = \nu_2$, and $\nu_1(x) \in I$ holds.*
  - **Assign** *$\varphi = x \leftarrow I$, then $w_1 = w_2$, $\nu_2 = \nu_1[x \leftarrow r]$ where $r \in I$.*
  - **Value passing** *$\varphi = x \leftarrow x'$, then $w_1 = w_2$, $\nu_2 = \nu_1[x \leftarrow \nu_1(x')]$.*
  - **Push** *$\varphi = push(\gamma)$, then $\nu_2 = \nu_{\mathbf{0}}$, $w_2 = (\gamma, (\nu_1(x_1), \cdots, \nu_k(x_k)), 1).w_1$ for $X = \{x_1, \cdots, x_k\}$.*
  - **F-Push** *$\varphi = fpush(\gamma)$, then $\nu_2 = \nu_{\mathbf{0}}$, $w_2 = (\gamma, (\nu_1(x_1), \cdots, \nu_k(x_k)), 0).w_1$ for $X = \{x_1, \cdots, x_k\}$.*
  - **Pop** *$\varphi = pop(\gamma)$, then $\nu_2 = \nu_1[\bar{x} \leftarrow (t_1, \cdots, t_k)]$, $w_1 = (\gamma, (t_1, \cdots, t_k), flag).w_2$.*

The initial configuration $\varrho_0 = (s_0, \epsilon, \nu_{\mathbf{0}})$. We use $\hookrightarrow$ to range over these transitions, and $\hookrightarrow^*$ is the reflexive and transitive closure of $\hookrightarrow$.

Example 1. The figure shows transitions $\varrho_1 \hookrightarrow \varrho_2 \hookrightarrow \varrho_3 \hookrightarrow \varrho_4$ of a DTPDA-F with $S = \{\bullet\}$ (omitted in the figure), $X = \{x_1, x_2\}$, $C = \{c_1\}$, and $\Gamma = \{a, b, d\}$. At $\varrho_1 \hookrightarrow \varrho_2$, the values of x_1 and x_2 (0.5 and 3.9) are pushed with d, and frozen. After pushing, value of x_1 and x_2 will be reset to zero, Then, x_2 is set a value in $(1, 2]$, say 1.7. At $\varrho_2 \hookrightarrow \varrho_3$, time elapses 2.6, but frozen ages in the top and third stack frames do not change. The rest (in **bold**) proceed. At $\varrho_3 \hookrightarrow \varrho_4$, test whether the value of x_2 is in $(4, 6)$. *Yes*, then pop the stack and x_1, x_2 are set to the popped ages. Last, the value of x_1 is set to c_1.



3 Reachability of DTPDAs with Frozen Ages

In this section, we assume $|C| = 1$, i.e., a DTPDA-F has a single global clock. We denote the set of finite multisets over D by $\mathcal{MP}(D)$, and the union of two multisets M, M' by $M \uplus M'$. We regard a finite set as a multiset with the multiplicity 1, and a finite word as a multiset by ignoring the ordering. We denote the top symbol and its suffix of a word w by $hd(w)$ and $tl(w)$, respectively.

3.1 Digiword and its Operations

Let $\langle S, s_0, \Gamma, X, C, \Delta\rangle$ be a DTPDA-F, and let n be the largest integer (except for ω) appearing in Δ. For $v \in \mathbb{R}^{\geq 0}$, $proj(v) = \mathtt{r}_i$ if $v \in \mathtt{r}_i \in Intv(n)$, where

$$Intv(n) = \{\mathtt{r}_{2i} = [i,i] \mid 0 \leq i \leq n\} \cup \{\mathtt{r}_{2i+1} = (i, i+1) \mid 0 \leq i < n\} \cup \{\mathtt{r}_{2n+1} = (n, \omega)\}$$

The idea of the next digitization is inspired by [13–15].

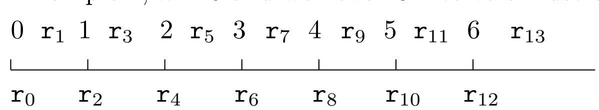
Definition 5. *Let $frac(x,t) = t - floor(t)$ for $(x,t) \in (C \cup X \cup \Gamma) \times \mathbb{R}^{\geq 0}$. A* digitization $\mathtt{digi} : \mathcal{MP}((C \cup X \cup \Gamma) \times \mathbb{R}^{\geq 0} \times \{0,1\}) \to \mathcal{MP}((C \cup X \cup \Gamma) \times Intv(n) \times \{0,1\})^*$ *is defined as follows.*

For $\bar{\mathcal{Y}} \in \mathcal{MP}((C \cup X \cup \Gamma) \times \mathbb{R}_{\geq 0} \times \{0,1\})$, let $Y_0, Y_1, \cdots, Y_m$ be multisets that collect $(x, proj(t), flag)$'s having the same $frac(x,t)$ for $(x,t,flag) \in \bar{\mathcal{Y}}$. Among them, Y_0 (which is possibly empty) is reserved for the collection of $(x, proj(t), flag)$ with $frac(t) = 0$ and $t \leq n$ (i.e., $proj(t) = \mathtt{r}_{2i}$ for $0 \leq i \leq n$). We assume that Y_i's except for Y_0 is non-empty (i.e., $Y_i = \emptyset$ with $i > 0$ is omitted), and Y_i's are sorted by the increasing order of $frac(x,t)$ (i.e., $frac(x,t) < frac(x',t')$ for $(x, proj(t), flag) \in Y_i$ and $(x', proj(t'), flag') \in Y_{i+1}$).

Note that $flag$ in $(x, proj(t), flag)$ is always 1 for $x \in C \cup X$. For $Y \in \mathcal{MP}((C \cup X \cup \Gamma) \times Intv(n) \times \{0,1\})$, we define the projections by $prc(Y) = \{(x, proj(t), 1) \in Y\}$ and $frz(Y) = \{(x, proj(t), 0) \in Y\}$. We overload the projections on $\bar{Y} = Y_0 Y_1 \cdots Y_m \in (\mathcal{MP}((C \cup X \cup \Gamma) \times Intv(n) \times \{0,1\}))^*$ such that $frz(\bar{Y}) = frz(Y_0) frz(Y_1) \cdots frz(Y_m)$ and $prc(\bar{Y}) = prc(Y_0) prc(Y_1) \cdots prc(Y_m)$.

For a stack frame $v = (\gamma, (t_1, \cdots, t_k), flag)$ of a DTPDA-F, we denote a word $(\gamma, t_1, flag) \cdots (\gamma, t_k, flag)$ by $dist(v)$. Given a clock valuation ν, we denote a clock word $(x_1, \nu(x_1), flag) \ldots (x_n, \nu(x_n), flag)$ where $x_1 \ldots x_n \in X \cup C$.

Example 2. In Example 1, $n = 6$ and we have 13 intervals illustrated below.



For the configuration $\varrho_1 = (\bullet, v_4 \cdots v_1, \nu)$ in Example 1, let $\bar{\mathcal{Y}} = dist(v_4) \uplus \ldots \uplus dist(v_1) \uplus time(\nu)$ be a word, and $\bar{Y} = \mathtt{digi}(\bar{\mathcal{Y}})$, i.e.,

$$\begin{aligned}
\bar{\mathcal{Y}} &= \{(a,1.9,1),(a,4.5,1),(b,6.7,0),(b,2.9,0),(a,3.1,1),(a,5.2,1),(d,4.2,1),\\
&\quad (d,3.3,1),(x_1,0.5,1),(x_2,3.9,1),(c_1,2.3,1)\}\\
\bar{Y} &= \{(a,\mathtt{r}_7,1)\}\{(a,\mathtt{r}_{11},1),(d,\mathtt{r}_9,1)\}\{(c_1,\mathtt{r}_5,1),(d,\mathtt{r}_7,1)\}\{(x_1,\mathtt{r}_1,1),(a,\mathtt{r}_9,1)\}\\
&\quad \{(b,\mathtt{r}_{13},0)\}\{(x_2,\mathtt{r}_7,1),(a,\mathtt{r}_3,1),(b,\mathtt{r}_5,0)\}\\
prc(\bar{Y}) &= \{(a,\mathtt{r}_7,1)\}\{(a,\mathtt{r}_{11},1),(d,\mathtt{r}_9,1)\}\{(c_1,\mathtt{r}_5,1),(d,\mathtt{r}_7,1)\}\{(x_1,\mathtt{r}_1,1),(a,\mathtt{r}_9,1)\}\\
&\quad \{(x_2,\mathtt{r}_7,1),(a,\mathtt{r}_3,1)\}\\
frz(\bar{Y}) &= \{(b,\mathtt{r}_{13},0)\}\{(b,\mathtt{r}_5,0)\}
\end{aligned}$$

A word in $(\mathcal{MP}((C \cup X \cup \Gamma) \times Intv(n) \times \{0,1\}))^*$ is called a *digiword*. We denote $\bar{Y}|_\Lambda$ for $\Lambda \subseteq \Gamma \cup C \cup X$, by removing $(x, \mathtt{r}_i, flag)$ with $x \notin \Lambda$. A *k-pointer* $\bar{\rho}$ of $\bar{Y}$ is a tuple of k pointers to mutually different k elements in $\bar{Y}|_\Gamma$. We refer the element pointed by the i-th pointer by $\bar{\rho}[i]$. From now on, we assume that

- the occurrence of $(x, \mathtt{r}_i, 1)$ with $x \in C \cup X$ in $\bar{Y}$ is exactly once, and
- a digiword has two pairs of k-pointers $(\bar{\rho}_1, \bar{\rho}_2)$ and $(\bar{\tau}_1, \bar{\tau}_2)$ that point to only proceeding and frozen ages, respectively. We call $(\bar{\rho}_1, \bar{\rho}_2)$ *proceeding* k-pointers and $(\bar{\tau}_1, \bar{\tau}_2)$ *frozen* k-pointers. We assume that they do not overlap each other, i.e., there are no i, j, such that $\bar{\rho}_1[i] = \bar{\rho}_2[j]$ or $\bar{\tau}_1[i] = \bar{\tau}_2[j]$.

$\bar{\rho}_1$ and $\bar{\rho}_2$ intend the store of values of the local clocks at the last and one before the last **Push**, respectively. $\bar{\tau}_1$ and $\bar{\tau}_2$ intend similar for **F-Push**.

Example 3. $\bar{Y}$ in Example 2 have proceeding 2-pointers $(\bar{\rho}_1, \bar{\rho}_2)$ (marked with the numbered overlines and underlines) frozen 2-pointers 2-pointers $(\bar{\tau}_1, \bar{\tau}_2)$ (marked with the numbered double overlines and double underlines).

$$\bar{Y} = \{\underline{(a, \mathtt{r}_7, 1)}_1\}\{\underline{(a, \mathtt{r}_{11}, 1)}_2, (d, \mathtt{r}_9, 1)\}\{(c_1, \mathtt{r}_5, 1), (d, \mathtt{r}_7, 1)\}\{(x_1, \mathtt{r}_1, 1), \overline{(a, \mathtt{r}_9, 1)}^2\}$$
$$\{\overline{\overline{(b, \mathtt{r}_{13}, 0)}}^1\}\{(x_2, \mathtt{r}_7, 1), \overline{(a, \mathtt{r}_3, 1)}^1, \overline{\overline{(b, \mathtt{r}_5, 0)}}^2\}$$
$$\bar{Y}|_\Gamma = \{\underline{(a, \mathtt{r}_7, 1)}_1\}\{\underline{(a, \mathtt{r}_{11}, 1)}_2, (d, \mathtt{r}_9, 1)\}\{(d, \mathtt{r}_7, 1)\}\{\overline{(a, \mathtt{r}_9, 1)}^2\}$$
$$\{\overline{\overline{(b, \mathtt{r}_{13}, 0)}}^1\}\{\overline{(a, \mathtt{r}_3, 1)}^1, \overline{\overline{(b, \mathtt{r}_5, 0)}}^2\}$$

Definition 6. *For digiwords $\bar{Y} = Y_1 \cdots Y_m$ and $\bar{Z} = Z_1 \cdots Z_{m'}$ with pairs of k-pointers $(\bar{\rho}_1, \bar{\rho}_2), (\bar{\tau}_1, \bar{\tau}_2)$, and $(\bar{\rho}'_1, \bar{\rho}'_2), (\bar{\tau}'_1, \bar{\tau}'_2)$, respectively. We define an embedding $\bar{Y} \sqsubseteq \bar{Z}$, if there exists a monotonic injection $f : [1..m] \to [1..m']$ such that $Y_i \subseteq Z_{f(i)}$ for each $i \in [1..m]$, $f \circ \bar{\rho}_i = \bar{\rho}'_i$ and $f \circ \bar{\tau}_i = \bar{\tau}'_i$ for $i = 1, 2$.*

Definition 7. *Let $\bar{Y} = Y_0 \cdots Y_m, \bar{Y}' = Y'_0 \cdots Y'_{m'} \in (\mathcal{MP}((\Gamma \cup C \cup X) \times Intv(n) \times \{0,1\}))^*$ such that $\bar{Y}$ (resp. $\bar{Y}'$) has two pairs of proceeding and frozen k-pointers $(\bar{\rho}_1, \bar{\rho}_2)$ and $(\bar{\tau}_1, \bar{\tau}_2)$ (resp. $(\bar{\rho}'_1, \bar{\rho}'_2)$ and $(\bar{\tau}'_1, \bar{\tau}'_2)$). We define digiword operations as follows. Note that except for* $\mathbf{Map}^{flag}_{\rightarrow}$, $\mathbf{Map}^{flag}_{\leftarrow}$, *and* **Permutation**, *k-pointers do not change.*

- **Decomposition** *Let $Z \in \mathcal{MP}((C \cup X \cup \Gamma) \times Intv(n) \times \{0,1\})$. If $Z \subseteq Y_j$, $decomp(\bar{Y}, Z) = (Y_0 \cdots Y_{j-1}, Y_j, Y_{j+1} \cdots Y_m)$.*
- $\mathbf{Insert}_I$ *Let $Z \in \mathcal{MP}((\Gamma \cup C \cup X) \times Intv(n) \times \{0,1\})$ with $(x, \mathtt{r}_i, flag) \in Z$ for $x \in C \cup X \cup \Gamma$. $insert_I(\bar{Y}, Z)$ inserts Z to $\bar{Y}$ such that*

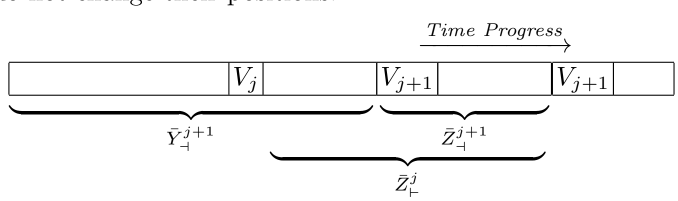
$$\begin{cases} \textit{either take the union of } Z \textit{ and } Y_j \textit{ for } j > 0, \textit{ or put } Z \textit{ at any place after } Y_0 \\ \qquad\qquad\qquad\qquad\qquad\qquad \textit{if } i \textit{ is odd} \\ \textit{take the union of } Z \textit{ and } Y_0 \qquad \textit{if } i \textit{ is even} \end{cases}$$

- $\mathbf{Insert}_x$ *$insert_x(\bar{Y}, x, y)$ adds $(x, \mathtt{r}_i, 1)$ to X_j for $(y, \mathtt{r}_i, 1) \in X_j$, $x, y \in C \cup X$.*
- **Init** *For $\bar{Y} = Y_0 \cdots Y_m$, $init(\bar{Y})$ is obtained by removing all elements $(x, \mathtt{r}_i)$ for $x \in X$ and updating Y_0 with $Y_0 \uplus \{(x_i, \mathtt{r}_0) \mid x_i \in X\}$.*
- **Delete** *$delete(\bar{Y}, x)$ for $x \in C \cup X$ is obtained from $\bar{Y}$ by deleting the element $(x, \mathtt{r})$ indexed by x.*

- **Permutation**. *Let* $\bar{V} = prc(\bar{Y}) = V_0 V_1 \cdots V_k$ *and* $\bar{U} = frz(\bar{Y}) = U_0 U_1 \cdots U_{k'}$. *A one-step permutation* $\bar{Y} \Rightarrow \bar{Y}'$ *is given by* $\Rightarrow \ = \ \Rightarrow_s \cup \Rightarrow_c$, *defined below. We denote* $inc(V_j)$ *for* V_j *in which each* $\mathtt{r}_i$ *is updated to* $\mathtt{r}_{i+1}$ *for* $i < 2k+1$.
  - **($\Rightarrow_s$)** *Let*
    $$\begin{cases} decomp(U_0\ inc(V_0)\ .\ tl(\bar{Y}),\ V_k) = (\bar{Y}^k_{\dashv},\ \hat{Y}^k,\ \bar{Y}^k_{\vdash}) \\ decomp(insert_I((\hat{Y}^k \setminus V_k)\ .\ \bar{Y}^k_{\vdash},\ V_k),\ V_k) = (\bar{Z}^k_{\dashv},\ \hat{Z}^k,\ \bar{Z}^k_{\vdash}). \end{cases}$$
    For j *with* $0 \leq j < k$, *we repeat to set*
    $$\begin{cases} decomp(\bar{Y}^{j+1}_{\dashv}\ .\ \bar{Z}^{j+1}_{\dashv},\ V_j) = (\bar{Y}^j_{\dashv},\ \hat{Y}^j,\ \bar{Y}^j_{\vdash}) \\ decomp(insert_I((\hat{Y}^j \setminus V_j)\ .\ \bar{Y}^j_{\vdash},\ V_j),\ V_j) = (\bar{Z}^j_{\dashv},\ \hat{Z}^j,\ \bar{Z}^j_{\vdash}). \end{cases}$$
    Then, $\bar{Y} \Rightarrow_s \bar{Y}' = \bar{Y}^0_{\dashv}\ \bar{Z}^0_{\dashv}\ \hat{Z}^0\ \bar{Z}^1_{\dashv}\ \hat{Z}^1 \cdots \bar{Z}^k_{\dashv}\ \hat{Z}^k\ \bar{Z}^k_{\vdash}$.
  - **($\Rightarrow_c$)** *Let* $\bar{Y}^k_{\dashv} = U_0 \cup inc(V_k)$ *and* $\bar{Z}^k_{\dashv} = inc(V_0)\ Y_1 \cdots (Y_{i'} \setminus V_k) \cdots Y_m$.
    For j *with* $0 \leq j < k$, *we repeat to set*
    $$\begin{cases} decomp(\bar{Y}^{j+1}_{\dashv}.\bar{Z}^{j+1}_{\dashv}, V_j) = (\bar{Y}^j_{\dashv}, \hat{Y}^j, \bar{Y}^j_{\vdash}) \\ decomp(insert_I((\hat{Y}^j \setminus V_j).\bar{Y}^j_{\vdash}, V_j), V_j) = (\bar{Z}^j_{\dashv}, \hat{Z}^j, \bar{Z}^j_{\vdash}). \end{cases}$$
    Then, $\bar{Y} \Rightarrow_c \bar{Y}' = \bar{Y}^0_{\dashv}\ \bar{Z}^0_{\dashv}\ \hat{Z}^0\ \bar{Z}^1_{\dashv}\ \hat{Z}^1 \cdots \bar{Z}^{k-1}_{\dashv}\ \hat{Z}^{k-1}\ \bar{Z}^{k-1}_{\vdash}$.

  $(\bar{\rho}_1, \bar{\rho}_2)$ *is updated to correspond to the permutation accordingly, and* $(\bar{\tau}_1, \bar{\tau}_2)$ *is kept unchanged.*
- **Rotate** *For proceeding k-pointers* $(\bar{\rho}_1, \bar{\rho}_2)$ *of* $\bar{Y}$ *and* $\bar{\rho}'$ *of* $\bar{Z}$, *let* $\bar{Y}|_{\Gamma} \Rightarrow^* \bar{Z}|_{\Gamma}$ *such that the permutation makes* $\bar{\rho}_1$ *match with* $\bar{\rho}$. *Then,* $rotate_{\bar{\rho}_1 \mapsto \bar{\rho}}(\bar{\rho}_2)$ *is the corresponding k-pointer of* $\bar{Z}$ *to* $\bar{\rho}_2$.
- $\mathbf{Map}^{flag}_{\rightarrow}$ $map^{fl}_{\rightarrow}(\bar{Y}, \gamma)$ *for* $\gamma \in \Gamma$ *is obtained from* $\bar{Y}$ *by, for each* $x_i \in X$, *replacing* $(x_i, \mathtt{r}_j, 1)$ *with* $(\gamma, \mathtt{r}_j, fl)$. *Accordingly, if* $fl = 1$, $\bar{\rho}_1[i]$ *is updated to point to* $(\gamma, \mathtt{r}_j, 1)$, *and* $\bar{\rho}_2$ *is set to the original* $\bar{\rho}_1$. *If* $fl = 0$, $\bar{\tau}_1[i]$ *is updated to point to* $(\gamma, \mathtt{r}_j, 0)$, *and* $\bar{\tau}_2$ *is set to the original* $\bar{\tau}_1$.
- $\mathbf{Map}^{flag}_{\leftarrow}$ $map^{fl}_{\leftarrow}(\bar{Y}, \bar{Y}', \gamma,)$ *for* $\gamma \in \Gamma$ *is obtained,*
  - **(if** $fl = 1$**)** *by replacing each* $\bar{\rho}_1[i] = (\gamma, \mathtt{r}_j, 1)$ *in* $\bar{Y}|_{C \cup \Gamma}$ *with* $(x_i, \mathtt{r}_j, 1)$ *for* $x_i \in X$. *Accordingly, new* $\bar{\rho}_1$ *is set to the original* $\bar{\rho}_2$, *and new* $\bar{\rho}_2$ *is set to* $rotate_{\bar{\rho}'_1 \mapsto \bar{\rho}_2}(\bar{\rho}'_2)$. $\bar{\tau}_1$ *and* $\bar{\tau}_2$ *are kept unchanged.*
  - **(if** $fl = 0$**)** *by replacing each* $\bar{\tau}_1[i] = (\gamma, \mathtt{r}_j, 0)$ *in* $\bar{Y}|_{C \cup \Gamma}$ *with* $(x_i, \mathtt{r}_j, 1)$ *for* $x_i \in X$. *Accordingly, new* $\bar{\tau}_1$ *is set to the original* $\bar{\tau}_2$, *and new* $\bar{\rho}_2$ *is set to* $\bar{\rho}'_2$. $\bar{\rho}_1$ *and* $\bar{\rho}_2$ *are kept unchanged.*

Remark 2. **Permutation** intends to describe (nondeterministic) time progress. The figure shows that, after where V_{j+1} shifts is decided, $\bar{Y}^{j+1}_{\dashv}.\bar{Z}^{j+1}_{\dashv}$ describes the prefix of the destination of V_{j+1}. Then, the possible destination of V_j is in $\bar{Y}^{j+1}_{\dashv}.\bar{Z}^{j+1}_{\dashv}$ after the current occurrence of V_j. This range is denoted by $\bar{Z}^j_{\vdash}$. Note that U_i's do not change their positions.



Example 4. We begin with the digiword $\bar{Y}$ in Example 3, to simulate transitions $\varrho_1 \hookrightarrow^* \varrho_3$ in Example 1.

- $fpush(d)$ is simulated by $\bar{Y}_1 = init(map^1_{\rightarrow}(\bar{Y}, \gamma))$.
  $\bar{Y}_1 = \{(x_1, \mathtt{r}_0, 1), (x_2, \mathtt{r}_0, 1)\}\{\underline{(a, \mathtt{r}_7, 1)}_1\}\{\underline{(a, \mathtt{r}_{11}, 1)}_2, (d, \mathtt{r}_9, 1)\}\{(c_1, \mathtt{r}_5, 1), (d, \mathtt{r}_7, 1)\}$
  $\{\overline{(a, \mathtt{r}_9, 1)}^2, \overline{\overline{(d, \mathtt{r}_1, 0)}}^1\}\{\underline{\underline{(b, \mathtt{r}_{13}, 0)}}_1\}\{\overline{(a, \mathtt{r}_3, 1)}^1, \underline{\underline{(b, \mathtt{r}_5, 0)}}_2, \overline{\overline{(d, \mathtt{r}_7, 0)}}^1\}$
- $x_2 \leftarrow (1, 2]$ is simulated by $\bar{Y}_2 = insert_I(delete(\bar{Y}_1, x_2), (x_2, \mathtt{r}_i)$.
  $\bar{Y}_2 = \{(x_1, \mathtt{r}_0, 1)\}\{\underline{(a, \mathtt{r}_7, 1)}_1\}\{\underline{(a, \mathtt{r}_{11}, 1)}_2, (d, \mathtt{r}_9, 1)\}\{(c_1, \mathtt{r}_5, 1), (d, \mathtt{r}_7, 1)\}$
  $\{\overline{(a, \mathtt{r}_9, 1)}^2, \overline{\overline{(d, \mathtt{r}_1, 0)}}^1\}\{(x_2, \mathtt{r}_3, 1), \underline{\underline{(b, \mathtt{r}_{13}, 0)}}_1\}\{\overline{(a, \mathtt{r}_3, 1)}^1, \underline{\underline{(b, \mathtt{r}_5, 0)}}_2, \overline{\overline{(d, \mathtt{r}_7, 0)}}^1\}$
- Time elapse of 2.6 time units is simulated by $\bar{Y}_2 \Rightarrow^* \bar{Y}_3$
  $\bar{Y}_3 = \{\overline{(a, \mathtt{r}_{13}, 1)}^2\}\{(x_2, \mathtt{r}_9, 1)\}\{\overline{(a, \mathtt{r}_9, 1)}^1, \overline{\overline{(d, \mathtt{r}_1, 0)}}^1\}\{(x_1, \mathtt{r}_5, 1)\}\{\underline{(a, \mathtt{r}_{11}, 1)}_1,$
  $\underline{\underline{(b, \mathtt{r}_{13}, 0)}}_1\}\{\underline{(a, \mathtt{r}_{13}, 1)}_2, (d, \mathtt{r}_{13}, 1)\}\{(c_1, \mathtt{r}_9, 1), (d, \mathtt{r}_{11}, 1), \underline{\underline{(b, \mathtt{r}_5, 0)}}_2, \overline{\overline{(d, \mathtt{r}_7, 0)}}^1\}$

3.2 Snapshot Pushdown System

A *snapshot pushdown system* (snapshot PDS) keeps the digitization of all values of (global and local) clocks and ages in the top stack frame, as a *digiword*. It is associated with a flag, which shows that the last push is either **Push** ($flag = 1$) or **F-Push** ($flag = 0$). It contains both proceeding and frozen ages, and only proceeding ages proceed synchronously to global and local clocks.

We show that a DTPDA-F with a single global clock is encoded into its digitization, called a *snapshot PDS*. The keys of the encoding are, (1) when a pop occurs, the time progress recorded at the top stack symbol is propagated to the next stack symbol after finding a permutation by matching between proceeding k-pointers $\bar{\rho}_2$ and $\bar{\rho}'_1$, and (2) the single global clock assumption allows us to compare current local clock values with a past one (which is stored in the global clock), but unable to compare past local clock values.

Definition 8. *Let $\pi : \varrho_0 = (q_0, \epsilon, \nu_{\mathbf{0}}) \hookrightarrow^* \varrho = (s, w, \nu)$ be a transition sequence of a DTPDA-F from the initial configuration. If π is not empty, we refer the last step as $\lambda : \varrho' \hookrightarrow \varrho$, and the preceding sequence by $\pi' : \varrho_0 \hookrightarrow^* \varrho'$. Let $w = v_m \cdots v_1$. A* snapshot *is $snap(\pi) = (\bar{Y}, flag(v_m))$, where*

$$\bar{Y} = \mathtt{digi}(\uplus_i dist(v_i) \uplus \{(x, \nu(x), 1) \mid x \in C \cup X\})$$

Let a k-pointer $\bar{\xi}(\pi)$ be $\bar{\xi}(\pi)[i] = (\gamma, proj(t_i), flag(v_m))$ for $(\gamma, t_i) \in dist(v_m)$. A snapshot configuration *$Snap(\pi)$ is inductively defined from $Snap(\pi')$.*

$$\begin{cases} (q_0, snap(\epsilon)) & \text{if } \pi = \epsilon.\ (\bar{\rho}_1, \bar{\rho}_2) \text{ and } (\bar{\tau}_1, \bar{\tau}_2) \text{ are undefined.} \\ (s', snap(\pi)\ tail(Snap(\pi'))) & \text{if } \lambda \text{ is } \textbf{Time progress} \text{ with } \bar{Y}' \Rightarrow^* \bar{Y}. \\ & \text{Then, the permutation } \bar{Y}' \Rightarrow^* \bar{Y} \text{ updates } (\bar{\rho}'_1, \bar{\rho}'_2) \text{ to } (\bar{\rho}_1, \bar{\rho}_2). \\ (s', snap(\pi)\ tail(Snap(\pi'))) & \text{if } \lambda \text{ is } \textbf{Local}, \textbf{Test}, \textbf{Assign}, \textbf{Value-passing}. \\ (s, snap(\pi)\ Snap(\pi')) & \text{if } \lambda \text{ is } \textbf{Push}. \text{ Then, } (\bar{\rho}_1, \bar{\rho}_2) = (\bar{\xi}(\pi), \bar{\rho}'_1). \\ (s, snap(\pi)\ Snap(\pi')) & \text{if } \lambda \text{ is } \textbf{F-Push}. \text{ Then, } (\bar{\tau}_1, \bar{\tau}_2) = (\bar{\xi}(\pi), \bar{\tau}'_1). \\ (s, snap(\pi)\ tail(tail(Snap(\pi')))) & \text{if } \lambda \text{ is } \textbf{Pop}. \\ \quad \text{If } flag = 1,\ (\bar{\rho}_1, \bar{\rho}_2) = (\bar{\rho}'_2, rotate_{\bar{\rho}''_1 \mapsto \bar{\rho}'_2}(\bar{\rho}''_2)); \text{ otherwise, } (\bar{\tau}_1, \bar{\tau}_2) = (\bar{\tau}'_2, \bar{\tau}''_2). & \end{cases}$$

We refer $head(Snap(\pi'))$ by $(\bar{Y}', flag')$, $head(tail(Snap(\pi'))$ by $(\bar{Y}'', flag'')$. Pairs of proceeding k-pointers of $\bar{Y}$, $\bar{Y}'$, and $\bar{Y}''$ are denoted by $(\bar{\rho}_1, \bar{\rho}_2)$, $(\bar{\rho}'_1, \bar{\rho}'_2)$, and $(\bar{\rho}''_1, \bar{\rho}''_2)$, respectively. Similarly, pairs of frozen ones are denoted by $(\bar{\tau}_1, \bar{\tau}_2)$, $(\bar{\tau}'_1, \bar{\tau}'_2)$, and $(\bar{\tau}''_1, \bar{\tau}''_2)$, respectively. If not mentioned, k-pointers are kept as is.

Example 5. In Example 1, ϱ_3 is described by $Snap(\pi)$ below for an execution path $\pi = \cdots \hookrightarrow \varrho_1 \hookrightarrow \varrho_2 \hookrightarrow \varrho_3$ from the initial configuration to ϱ_3.

$(\{\overline{(a,\mathtt{r}_{13},1)}^{2}\}\{(x_2,\mathtt{r}_9,1)\}\{\overline{(a,\mathtt{r}_9,1)}^{1}, \overline{\overline{(d,\mathtt{r}_1,0)}}^{1}\}\{(x_1,\mathtt{r}_5,1)\}\{\underline{(a,\mathtt{r}_{11},1)}_{1}, \underline{\underline{(b,\mathtt{r}_{13},0)}}_{1}\}$ $\{\underline{(a,\mathtt{r}_{13},1)}_{2}, (d,\mathtt{r}_{13},1)\}\{(c_1,\mathtt{r}_9,1),(d,\mathtt{r}_{11},1), \underline{\underline{(b,\mathtt{r}_5,0)}}_{2}, \overline{\overline{(d,\mathtt{r}_7,0)}}^{2}\}, \quad fl=0)$
$(\{\underline{(a,\mathtt{r}_7,1)}_{1}\}\{\underline{(a,\mathtt{r}_{11},1)}_{2}, (d,\mathtt{r}_9,1)\}\{(c_1,\mathtt{r}_5,1),(d,\mathtt{r}_7,1)\}\{(x_1,\mathtt{r}_1,1), \overline{(a,\mathtt{r}_9,1)}^{2}\}$ $\{\overline{\overline{(b,\mathtt{r}_{13},0)}}^{1}\}\{(x_2,\mathtt{r}_7,1), \overline{(a,\mathtt{r}_3,1)}^{1}, \overline{\overline{(b,\mathtt{r}_5,0)}}^{2}\}, \quad fl=1)$
$(\{\overline{(a,\mathtt{r}_7,1)}^{1}\}\{\overline{(a,\mathtt{r}_{11},1)}^{2}, \underline{(d,\mathtt{r}_9,1)}_{1}\}\{(c_1,\mathtt{r}_5,1), \underline{(d,\mathtt{r}_7,1)}_{2}\}\{(x_1,\mathtt{r}_1,1)\}$ $\{\overline{\overline{(b,\mathtt{r}_{13},0)}}^{1}\}\{(x_2,\mathtt{r}_7,1), \overline{\overline{(b,\mathtt{r}_5,0)}}^{2}\}, \quad fl=0)$
$(\{\overline{(a,\mathtt{r}_7,1)}^{1}\}\{\overline{(a,\mathtt{r}_{11},1)}^{2}, \underline{(d,\mathtt{r}_9,1)}_{1}\}\{(c_1,\mathtt{r}_5,1), \underline{(d,\mathtt{r}_7,1)}_{2}\}\{(x_1,\mathtt{r}_1,1)\}\{(x_2,\mathtt{r}_7,1)\},$ $fl=1)$
$(\{\overline{(d,\mathtt{r}_9,1)}^{1}\}\{(c_1,\mathtt{r}_5,1), \overline{(d,\mathtt{r}_7,1)}^{2}\}\{(x_1,\mathtt{r}_1,1)\}\{(x_2,\mathtt{r}_7,1)\}, \quad fl=1)$

Definition 9. *For a DTPDA-F $\langle S, s_0, \Gamma, X, C, \Delta\rangle$ with $|C| = 1$, a* snapshot PDS $\mathcal{S}$ *is a PDS (ith possibly infinite stack alphabet)*

$$\langle S, s_0, (\mathcal{MP}((C \cup X \cup \Gamma) \times Intv(n) \times \{0,1\}))^*, \Delta_d\rangle.$$

with the initial configuration $\langle s_{init}, \{(x, \mathtt{r}_0) \mid x \in C \cup X\}\rangle$. Then Δ_d consists of:

Time progress $\langle s, (\bar{Y}, flag)\rangle \hookrightarrow_{\mathcal{S}} \langle s, (\bar{Y}',, flag)\rangle$ *for* $\bar{Y} \Rightarrow^* \bar{Y}'$.
Local *($s \xrightarrow{\epsilon} s' \in \Delta$)* $\langle s, (\bar{Y}, flag)\rangle \hookrightarrow_{\mathcal{S}} \langle s', (\bar{Y}, flag)\rangle$.
Test *($s \xrightarrow{x \in I?} s' \in \Delta$)* *If $\mathtt{r}_i \subseteq I$ and $(x, \mathtt{r}_i, 1) \in \bar{Y}$,*
$\langle s, (\bar{Y}, flag)\rangle \hookrightarrow_{\mathcal{S}} \langle s', (\bar{Y}, flag)\rangle$.
Assign *($s \xrightarrow{x \leftarrow I} s' \in \Delta$ with $x \in X$)* *For $\mathtt{r}_i \subseteq I$,*
$\langle s, (\bar{Y}, flag)\rangle \hookrightarrow_{\mathcal{S}} \langle s', (insert_I(delete(\bar{Y}, x), \{(x, \mathtt{r}_i, 1)\}), flag)\rangle$.
Assign *($s \xrightarrow{c \leftarrow I} s' \in \Delta$ with $c \in C$)* *For $\mathtt{r}_i \subseteq I$,*
$\langle s, (\bar{Y}, flag)\rangle \hookrightarrow_{\mathcal{S}} \langle s', (insert_I(delete(\bar{Y}, c), \{(c, \mathtt{r}_i, 1)\}), flag)\rangle$.
Value-passing *($s \xrightarrow{x \leftarrow y} s' \in \Delta$ with $x \in X$)*
$\langle s, (\bar{Y}, flag)\rangle \hookrightarrow_{\mathcal{S}} \langle s', (insert_x(delete(\bar{Y}, c), x, y), flag)\rangle$.
Value-passing *($s \xrightarrow{c \leftarrow y} s' \in \Delta$ with $c \in C$)*
$\langle s, (\bar{Y}, flag)\rangle \hookrightarrow_{\mathcal{S}} \langle s', (insert_x(delete(\bar{Y}, c), c, y), flag)\rangle$.
Push *($s \xrightarrow{push(\gamma)} s' \in \Delta$; $fl = 1$) and* **F-Push** *($s \xrightarrow{fpush(\gamma)} s' \in \Delta$; $fl = 0$)*
$\langle s, (\bar{Y}, flag)\rangle \hookrightarrow_{\mathcal{S}} \langle s', (init(map^{fl}_{\rightarrow}(\bar{Y}, \gamma)), fl)(\bar{Y}, flag)\rangle$.
Pop *($s \xrightarrow{pop(\gamma)} s' \in \Delta$)*
$\langle s, (\bar{Y}, flag)(\bar{Y}', flag')\rangle \hookrightarrow_{\mathcal{S}} \langle s', (map^{flag}_{\leftarrow}(\bar{Y}, \bar{Y}', \gamma), flag')\rangle$.

Example 6. Following to Example 5, $\varrho_3 \hookrightarrow \varrho_4$ in Example 1 is described by $Snap(\pi) \hookrightarrow_{\mathcal{S}} Snap(\pi')$ with $Snap(\pi')$ below for $\pi' = \pi \hookrightarrow \varrho_4$.

$(\; \{\overline{(a, \mathtt{r}_{13}, 1)}^{2}\}\{(x_2, \mathtt{r}_9, 1)\}\{\overline{(a, \mathtt{r}_9, 1)}^{1}\}\{\underline{(a, \mathtt{r}_{11}, 1)}_{1}, \overline{\overline{(b, \mathtt{r}_{13}, 0)}}^{1}\}$ $\{\underline{(a, \mathtt{r}_{13}, 1)}_{2}, (d, \mathtt{r}_{13}, 1)\}\{(x_1, \mathtt{r}_{11}, 1), (c_1, \mathtt{r}_9, 1), (d, \mathtt{r}_{11}, 1), \overline{\overline{(b, \mathtt{r}_5, 0)}}^{2}\}, \quad fl = 1 \;)$
$(\; \{\overline{(a, \mathtt{r}_7, 1)}^{1}\}\{\overline{(a, \mathtt{r}_{11}, 1)}^{2}, \underline{(d, \mathtt{r}_9, 1)}_{1}\}\{(c_1, \mathtt{r}_5, 1), \underline{(d, \mathtt{r}_7, 1)}_{2}\}\{(x_1, \mathtt{r}_1, 1)\}$ $(\; \{\overline{\overline{(b, \mathtt{r}_{13}, 0)}}^{1}\}\{(x_2, \mathtt{r}_7, 1), \overline{\overline{(b, \mathtt{r}_5, 0)}}^{2}\}, \quad fl = 0 \;)$
$(\; \{\overline{(a, \mathtt{r}_7, 1)}^{1}\}\{\overline{(a, \mathtt{r}_{11}, 1)}^{2}, \underline{(d, \mathtt{r}_9, 1)}_{1}\}\{(c_1, \mathtt{r}_5, 1), \underline{(d, \mathtt{r}_7, 1)}_{2}\}\{(x_1, \mathtt{r}_1, 1)\}\{(x_2, \mathtt{r}_7, 1)\},$ $fl = 1 \;)$
$(\; \{\overline{(d, \mathtt{r}_9, 1)}^{1}\}\{(c_1, \mathtt{r}_5, 1), \overline{(d, \mathtt{r}_7, 1)}^{2}\}\{(x_1, \mathtt{r}_1, 1)\}\{(x_2, \mathtt{r}_7, 1)\}, \; fl = 1 \;)$

By induction on the number of steps of transitions, the encoding relation between a DTPDA-F with a single global clock and a snapshot PDS is observed. Note that the initial clock valuation of the DTPDA-F to be set $\nu_\mathbf{0}$ is essential.

Lemma 1. *Let us denote ϱ_0 and ϱ (resp. $\langle q_0, \tilde{w}_0\rangle$ and $\langle s, \tilde{w}\rangle$) for the initial configuration and a configuration of a DTPDA-F (resp. its snapshot PDS $\mathcal{S}$).*

(Preservation) *If $\pi : \varrho_0 \hookrightarrow^* \varrho$, there exists $\langle s, \tilde{w}\rangle$ such that $\langle q_0, \tilde{w}_0\rangle \hookrightarrow^*_{\mathcal{S}} \langle s, \tilde{w}\rangle$ and $Snap(\pi) = \langle s, \tilde{w}\rangle$.*

(Reflection) *If $\langle q_0, \tilde{w}_0\rangle \hookrightarrow^*_{\mathcal{S}} \langle s, \tilde{w}\rangle$, there exists $\pi : \varrho_0 \hookrightarrow^* \varrho$ with $Snap(\pi) = \langle s, \tilde{w}\rangle$.*

3.3 Well-Formed Constraint

A snapshot PDS is *a growing WSPDS* (Definition 6 in [9]) and $\Downarrow_\Upsilon$ gives a *well-formed constraint* (Definition 8 in [9]). Let us recall the definitions.

Let P be a set of control locations and let Γ be a stack alphabet. Different from an ordinary definition of PDSs, we do not assume that P and Γ are finite, but associated with well-quasi-orderings (WQOs) $\preceq$ and $\leq$, respectively. Note that the embedding $\sqsubseteq$ over digiwords is a WQO by Higman's lemma.

For $w = \alpha_1\alpha_2 \cdots \alpha_n, v = \beta_1\beta_2 \cdots \beta_m \in \Gamma^*$, let $w \lll v$ if $m = n$ and $\forall i \in [1..n].\alpha_i \leq \beta_i$. We extend $\lll$ on configurations such that $(p, w) \lll (q, v)$ if $p \preceq q$ and $w \lll v$ for $p, q \in P$ and $w, v \in \Gamma^*$. A partial function $\psi \in \mathcal{P}Fun(X, Y)$ is *monotonic* if $\gamma \leq \gamma'$ with $\gamma \in dom(\psi)$ implies $\psi(\gamma) \lll \psi(\gamma')$ and $\gamma' \in dom(\psi)$.

A *a well-structured PDS* (WSPDS) is a triplet $\langle (P, \preceq), (\Gamma, \leq), \Delta\rangle$ of a set $(P, \preceq)$ of WQO states, a WQO stack alphabet $(\Gamma, \leq)$, and a finite set $\Delta \subseteq \mathcal{P}Fun(P \times \Gamma, P \times \Gamma^{\leq 2})$ of monotonic partial functions. A WSPDS is *growing* if, for each $\psi(p, \gamma) = (q, w)$ with $\psi \in \Delta$ and $(q', w') \ggg (q, w)$, there exists (p', γ') with $(p', \gamma') \ggg (p, \gamma)$ such that $\psi(p', \gamma') \ggg (q', w')$.

Definition 10. *For a WSPDS $\langle (P, \preceq), (\Gamma, \leq), \Delta\rangle$, a pair $(\Upsilon, \Downarrow_\Upsilon)$ of a set $\Upsilon \subseteq P \times \Gamma^*$ and a projection function $\Downarrow_\Upsilon$: $P \times \Gamma^* \to (P \times \Gamma^*) \cup \{\#\}$ is a* well-formed constraint *if, for configurations c, c',*

- *$c \hookrightarrow c'$ implies that $c \in \Upsilon$ if, and only if $c' \in \Upsilon$,*
- *$c \hookrightarrow c'$ implies $\Downarrow_\Upsilon (c) \hookrightarrow \Downarrow_\Upsilon (c')$,*
- *$\Downarrow_\Upsilon (c) \lll c$, and*
- *$c \lll c'$ implies either $\Downarrow_\Upsilon (c) = \Downarrow_\Upsilon (c')$ or $\Downarrow_\Upsilon (c) = \#$,*

where # is added to $P \times \Gamma^*$ *as the least element (wrt* $\leqslant$*) and* $\Upsilon = \{c \in P \times \Gamma^* \mid c = \Downarrow_\Upsilon (c)\}$*. (# represents failures of* $\Downarrow_\Upsilon$*.)*

A well-formed constraint describes a syntactical feature that is preserved under transitions. Theorem 3 in [9] ensures a P-automaton construction for the quasi-coverability of a growing WSPDS with *directed* WQOs[4], Theorem 4 in [9] ensures the finite convergence of a P-automaton, and Theorem 5 in [9] lifts it to the reachability when it has a well-formed constraint.

Definition 11. *Let a configuration* $(s, \tilde{w})$ *of a snapshot PDS* $\mathcal{S}$*. An element in a stack frame of* $\tilde{w}$ *has a* parent *if it has a corresponding element in the next stack frame. The transitive closure of the parent relation is* an ancestor*. An element in* $\tilde{w}$ *is* marked*, if its ancestor is pointed by a k-pointer in some stack frame. We define a* projection $\Downarrow_\Upsilon (\tilde{w})$ *by removing unmarked elements in* $\tilde{w}$*. We say that* $\tilde{w}$ *is* well-formed *if* $\Downarrow_\Upsilon (\tilde{w}) = \tilde{w}$*.*

The idea of $\Downarrow_\Upsilon$ is, to remove unnecessary elements (i.e., elements not related to previous actions) from the stack content. Note that a configuration reachable from the initial configuration by $\hookrightarrow^*_{\mathcal{S}}$ is always well-formed. Since a snapshot PDS is a growing WSPDS with $\Downarrow_\Upsilon$, we conclude our first theorem from Lemma 1.

Theorem 1. *The reachability of a DTPDA-F* $\langle S, s_0, \Gamma, X, C, \Delta \rangle$ *is decidable, if* $|C| = 1$*.*

4 Nested Timed Automata with Frozen Clocks

4.1 Nested Timed Automata with Frozen Clocks

Definition 12 extends Definition 5 in [4] with the choice that all clocks of an interrupted TA are either proceeding or frozen. In [4], only the former is allowed. For simplicity, we assume that each $\mathcal{A}_i$ in T shares the same set of local clocks.

Definition 12 (Nested Timed Automata with Frozen Clocks). *A NeTA-F is a quadruplet* $\mathcal{N} = (T, \mathcal{A}_0, X, C, \Delta)$*, where*

- T *is a finite set* $\{\mathcal{A}_0, \mathcal{A}_1, \cdots, \mathcal{A}_k\}$ *of TAs, with the initial TA* $\mathcal{A}_0 \in T$*. We assume the sets of states of* $\mathcal{A}_i$*, denoted by* $S(\mathcal{A}_i)$*, are mutually disjoint, i.e.,* $S(\mathcal{A}_i) \cap S(\mathcal{A}_j) = \emptyset$ *for* $i \neq j$*. We denote the initial state of* $\mathcal{A}_i$ *by* $q_0(\mathcal{A}_i)$*.*
- C *is a finite set of global clocks, and* X *is the finite set of* k *local clocks.*
- $\Delta \subseteq Q \times (Q \cup \{\varepsilon\}) \times Actions^+ \times Q \times (Q \cup \{\varepsilon\})$ *describes transition rules below, where* $Q = \cup_{\mathcal{A}_i \in T} S(\mathcal{A}_i)$*.*

A transition rule is described by a sequence of $Actions = \{internal, push, fpush, pop, c \in I, c \leftarrow I, x \leftarrow c, c \leftarrow x\}$ *where* $c \in C$*,* $x \in X$*, and* $I \in \mathcal{I}$*. The internal actions are* **Local***,* **Test***,* **Assign***, and* **Value-passing** *in Definition 1.*

[4] The key Lemma 1 for Theorem 3 in [9] requires that a WQO is *directed* (i.e., for each x, y, there exists z with $z \geq x, y$), which was missing in [9].

Internal $(q, \varepsilon, internal, q', \varepsilon)$, *which describes an internal transition in the working TA (placed at a control location) with* $q, q' \in \mathcal{A}_i$.

Push $(q, \varepsilon, push, q_0(\mathcal{A}_{i'}), q)$, *which interrupts the currently working TA* $\mathcal{A}_i$ *at* $q \in S(\mathcal{A}_i)$. *Then, a TA* $\mathcal{A}_{i'}$ *newly starts. Note that all local clocks of* $\mathcal{A}_i$ *pushed onto the stack simultaneously proceed to global clocks.*

F-Push $(q, \varepsilon, fpush, q_0(\mathcal{A}_{i'}), q)$, *which is the same as* **Push** *except that all local clocks of* $\mathcal{A}_i$ *are frozen.*

Pop $(q, q', pop, q', \varepsilon)$, *which restarts* $\mathcal{A}_{i'}$ *in the stack from* $q' \in S(\mathcal{A}_{i'})$ *after* $\mathcal{A}_i$ *has finished at* $q \in S(\mathcal{A}_i)$.

Global-test $(q, \varepsilon, c \in I?, q, \varepsilon)$, *which tests whether the value of a global clock* c *is in* I.

Global-assign $(q, \varepsilon, c \leftarrow I, q, \varepsilon)$, *which assigns a value in* $r \in I$ *to a global clock* c.

Global-load $(q, \varepsilon, x \leftarrow c, q, \varepsilon)$, *which assign the value of a global clock* c *to a local clock* $x \in X$ *in the working TA.*

Global-store $(q, \varepsilon, c \leftarrow x, q, \varepsilon)$, *which assign the value of a local clock* $x \in X$ *of the working TA to a global clock* c.

Definition 13 (Semantics of NeTA-F). *Given a NeTA-F* $(T, \mathcal{A}_0, X, C, \Delta)$, *the current control state is referred by* q. *Let* $\mathcal{V}al_X = \{\nu : X \to \mathbb{R}^{\geq 0}\}$ *and* $\mathcal{V}al_C = \{\mu : C \to \mathbb{R}^{\geq 0}\}$. *A configuration of a NeTA-F is an element in* $(Q \times \mathcal{V}al_X \times \mathcal{V}al_C, (Q \times \{0,1\} \times \mathcal{V}al_X)^*)$.

- *Time progress transitions:* $(\langle q, \nu, \mu \rangle, v) \xrightarrow{t} (\langle q, \nu + t, \mu + t \rangle, v + t)$ *for* $t \in \mathbb{R}^{\geq 0}$, *where* $v + t$ *set* $\nu' := progress(\nu', t, flag)$ *of each* $\langle q', flag, \nu' \rangle$ *in the stack.*
- *Discrete transitions:* $\kappa \xrightarrow{\varphi} \kappa'$ *is defined as follows.*
  - **Internal** $(\langle q, \nu, \mu \rangle, v) \xrightarrow{\varphi} (\langle q', \nu', \mu \rangle, v)$, *if* $\langle q, \nu \rangle \xrightarrow{\varphi} \langle q', \nu' \rangle$ *is in Definition 2, except for* **push** *or* **pop**.
  - **Push** $(\langle q, \nu', \mu \rangle, v) \xrightarrow{push} (\langle q_0(\mathcal{A}_{i'}), \nu_{\mathbf{0}}, \mu \rangle, \langle q, 1, \nu \rangle.v)$.
  - **F-Push** $(\langle q, \nu', \mu \rangle, v) \xrightarrow{f\text{-}push} (\langle q_0(\mathcal{A}_{i'}), \nu_{\mathbf{0}}, \mu \rangle, \langle q, 0, \nu \rangle.v)$.
  - **Pop** $(\langle q, \nu, \mu \rangle, \langle q', flag, \nu' \rangle.w) \xrightarrow{pop} (\langle q', \nu', \mu \rangle, w)$.
  - **Global-test** $(\langle q, \nu, \mu \rangle, v) \xrightarrow{c \in I?} (\langle q, \nu, \mu \rangle, v)$, *if* $\mu(c) \in I$.
  - **Global-assign** $(\langle q, \nu, \mu \rangle, v) \xrightarrow{c \leftarrow I} (\langle q, \nu, \mu[c \leftarrow r] \rangle, v)$ *for* $r \in I$.
  - **Global-load** $(\langle q, \nu, \mu \rangle, v) \xrightarrow{x \leftarrow c} (\langle q, \nu[x \leftarrow \mu(c)], \mu \rangle, v)$.
  - **Global-store** $(\langle q, \nu, \mu \rangle, v) \xrightarrow{c \leftarrow x} (\langle q, \nu, \mu[c \leftarrow \nu(x)] \rangle, v)$.

The initial configuration of NeTA-F is $(\langle q_0(\mathcal{A}_0), \nu_{\mathbf{0}}, \mu_{\mathbf{0}} \rangle, \varepsilon)$, *where* $\nu_{\mathbf{0}}(x) = 0$ *for* $x \in X$ *and* $\mu_{\mathbf{0}}(c) = 0$ *for* $c \in C$. *We use* $\longrightarrow$ *to range over these transitions.*

4.2 Reachability of NeTA-Fs with Multiple Global Clocks

For showing the undecidability, we encode the halting problem of Minsky machines [16] in a NeTA-F. A Minsky machine $\mathcal{M}$ is a tuple (L, C, D) where:

- L is a finite set of states, and $l_f \in L$ is the terminal state,

– $C = \{ct_1, ct_2\}$ is the set of two counters, and
– D is the finite set of transition rules of the following types,
  - **increment counter** $d_i : ct := ct + 1$, goto l_k,
  - **test-and-decrement counter** d_i : if $(ct > 0)$ then $(ct := ct - 1$, goto $l_k)$ else goto l_m,

  where $ct \in C$, $d_i \in D$ and $l_k, l_m \in L$.

Example 7. By the N-wrapping technique [17], a Minsky machine can be encoded into a NeTA-F $\mathcal{N} = (T, \mathcal{A}_0, C, \Delta)$, with $T = \{\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2\}$ where

– $S(\mathcal{A}_0) = \{q_0\}$, $X(\mathcal{A}_0) = \{x_f, x_p\}$, $S(\mathcal{A}_1) = \{q_1\}$, $X(\mathcal{A}_1) = \{x_1, dum_1\}$, $S(\mathcal{A}_2) = \{q_2\}$, and $X(\mathcal{A}_2) = \{dum_2, dum_3\}$, where the dummy clocks dum_i's are prepared for fulfilling $k = 2$.
– $C = \{c_{sys}, c_v\}$ where c_{sys} is a system clock that will be reset to zero when its value becomes N; c_v encodes values of two counters as $\mu(c_v) = 2^{-ct_1} \cdot 3^{-ct_2}$.

Decrementing and incrementing the counter ct_1 are simulated by doubling and halving of the value of the clock c_v, respectively, while those for ct_2 are simulated by tripling and thirding the value of clock c_v. Zero-test of ct_1 is simulated by (1) multiplying the value of c_v by a power of 3, and (2) comparing it with 3. Similar for ct_2. These operations are illustrated in Fig. 1, and formally described below.

Doubling: Initially $\nu(c_{sys}) = 0$ and $\nu(c_v) = d$ with $0 < d < 1$. Then the doubling the value of c_v is obtained at the end, as $\nu(c_{sys}) = 0$ and $\nu(c_v) = 2d$.
$q_0 \xrightarrow{c_v \in [N,N]?} \xrightarrow{c_v \leftarrow [0,0]} q_0 \xrightarrow{c_{sys} \in [N,N]?} \xrightarrow{c_{sys} \leftarrow [0,0]} \xrightarrow{x_f \leftarrow c_v} \xrightarrow{f\text{-}push}$
$q_1\ q_0 \xrightarrow{c_v \in [N,N]?} \xrightarrow{pop} \xrightarrow{c_v \leftarrow x_f} q_0 \xrightarrow{c_{sys} \in [N,N]?} \xrightarrow{c_{sys} \leftarrow [0,0]} q_0$

Halving: During the halving the value of c_v, it will be nondeterministically stored to x_f in a frozen TA. When c_{sys} is reset to zero, x_f will be popped to restart. Only if the values of x_f and c_v coincide (i.e., they reach to N together), the value of c_v becomes $d/2$ when c_{sys} is wrapped twice.
$q_0 \xrightarrow{c_v \in [N,N]?} \xrightarrow{c_v \leftarrow [0,0]} q_0 \xrightarrow{x_f \leftarrow c_v} \xrightarrow{c_v \leftarrow [0,0]} \xrightarrow{f\text{-}push}$
$q_1\ q_0 \xrightarrow{c_{sys} \in [N,N]?} \xrightarrow{c_{sys} \leftarrow [0,0]} \xrightarrow{pop} q_0 \xrightarrow{c_v \in [N,N]?} \xrightarrow{x_f \in [N,N]?} \xrightarrow{c_v \leftarrow [0,0]}$
$q_0 \xrightarrow{c_{sys} \in [N,N]?} \xrightarrow{c_{sys} \leftarrow [0,0]} q_0$

Tripling: Tripling requires an extra local clock x_p in $\mathcal{A}_0$.
$q_0 \xrightarrow{c_v \in [N,N]?} \xrightarrow{c_v \leftarrow [0,0]} q_0 \xrightarrow{c_{sys} \in [N,N]?} \xrightarrow{c_{sys} \leftarrow [0,0]} \xrightarrow{x_f \leftarrow c_v} \xrightarrow{f\text{-}push}$
$q_1\ q_0 \xrightarrow{c_v \in [N,N]?} \xrightarrow{pop} \xrightarrow{c_v \leftarrow x_f} \xrightarrow{x_p \leftarrow [0,0]} q_0 \xrightarrow{c_{sys} \in [N,N]?} \xrightarrow{c_{sys} \leftarrow [0,0]} \xrightarrow{x_f \leftarrow c_v}$
$\xrightarrow{c_v \leftarrow x_p} \xrightarrow{f\text{-}push} q_1\ q_0 \xrightarrow{c_v \in [N,N]?} \xrightarrow{pop} \xrightarrow{c_v \leftarrow x_f} q_0 \xrightarrow{c_{sys} \in [N,N]?} \xrightarrow{c_{sys} \leftarrow [0,0]} q_0$

Thirding: Thirding requires an extra TA $\mathcal{A}_2$ with a local clock x_1 .
$q_0 \xrightarrow{c_v \in [N,N]?} \xrightarrow{c_v \leftarrow [0,0]} q_0 \xrightarrow{x_f \leftarrow c_v} \xrightarrow{c_v \leftarrow [0,0]} \xrightarrow{f\text{-}push}$
$q_1\ q_0 \xrightarrow{x_1 \leftarrow c_v} \xrightarrow{c_v \leftarrow [0,0]} \xrightarrow{f\text{-}push} q_2\ q_1\ q_0 \xrightarrow{c_{sys} \in [N,N]?} \xrightarrow{c_{sys} \leftarrow [0,0]} \xrightarrow{pop}$
$q_1\ q_0 \xrightarrow{c_v \in [N,N]?} \xrightarrow{x_1 \in [N,N]?} \xrightarrow{c_v \leftarrow [0,0]} q_1\ q_0 \xrightarrow{c_{sys} \in [N,N]?} \xrightarrow{c_{sys} \leftarrow [0,0]} \xrightarrow{pop}$
$q_0 \xrightarrow{c_v \in [N,N]?} \xrightarrow{x_f \in [N,N]?} \xrightarrow{c_v \leftarrow [0,0]} q_0 \xrightarrow{c_{sys} \in [N,N]?} \xrightarrow{c_{sys} \leftarrow [0,0]} q_0$

Theorem 2. *The reachability of a NeTA-F* $(T, \mathcal{A}_0, C, \Delta)$ *is undecidable, if* $|C| > 1$.

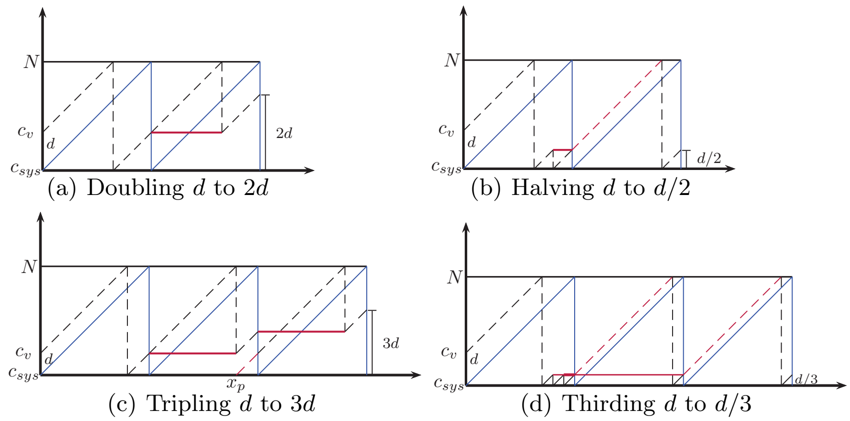


Fig. 1. Doubling, Halving, Tripling and Thirding the Value of c_v

4.3 Reachability of NeTA-F with a Single Global Clock

Let $\mathcal{N} = (T, \mathcal{A}_0, X, C, \Delta)$ be a NeTA-F. We define a corresponding DTPDA-F $\mathcal{E}(\mathcal{N}) = \langle S, s_0, \Gamma, X, C, \nabla \rangle$, such that

- $S = \Gamma = \bigcup_{\mathcal{A}_i \in T} S(\mathcal{A}_i)$ is the set of all locations of TAs in T, with
- $s_0 = q_0(\mathcal{A}_0)$ is the initial location of the initial TA $\mathcal{A}_0$ of $\mathcal{N}$.
- $X = \{x_1, \ldots, x_k\}$ is the set of k local clocks, and C is the singleton set $\{c\}$.
- ∇ is the union $\bigcup_{\mathcal{A}_i \in T} \Delta(\mathcal{A}_i) \bigcup \mathcal{G}(\mathcal{N}) \bigcup \mathcal{H}(\mathcal{N})$ where

$$\begin{cases} \Delta(\mathcal{A}_i) = \{\textbf{Local}, \textbf{Test}, \textbf{Assign}, \textbf{Value-passing}\}, \\ \mathcal{G}(\mathcal{N}) \ = \{\textbf{Global-test}, \textbf{Global-assign}, \textbf{Global-load}, \textbf{Global-store}\}, \\ \mathcal{H}(\mathcal{N}) \text{ consists of rules below.} \end{cases}$$

$$\begin{array}{lll} \textbf{Push} & q \xrightarrow{push(q)} q_0(\mathcal{A}_{i'}) & \text{if } (q, \varepsilon, push, q_0(\mathcal{A}_{i'}), q) \in \Delta(\mathcal{N}) \\ \textbf{F-Push} & q \xrightarrow{fpush(q)} q_0(\mathcal{A}_{i'}) & \text{if } (q, \varepsilon, f\text{-}push, q_0(\mathcal{A}_{i'}), q) \in \Delta(\mathcal{N}) \\ \textbf{Pop} & q \xrightarrow{pop(q')} q' & \text{if } (q, q', pop, q', \varepsilon)) \in \Delta(\mathcal{N}) \end{array}$$

Definition 14. *Let $\mathcal{N}$ be a NeTA-F $(T, \mathcal{A}_0, C, \Delta)$ and let $\mathcal{E}(\mathcal{N})$ be a DTPDA-F $\langle S, s_0, \Gamma, X, C, \nabla \rangle$. For a configuration $\kappa = (q, \nu, \mu\rangle, v)$ of $\mathcal{N}$ such that $v = (q_1, flag_1, \nu_1) \ldots (q_n, flag_n, \nu_n)$, $[\![\kappa]\!]$ denotes a configuration $(q, \overline{w}(\kappa), \nu \cup \mu)$ of $\mathcal{E}(\mathcal{N})$ where$q_i \in S(\mathcal{A}_i)$ and $\overline{w}(\kappa) = w_1 \cdots w_n$ with $w_i = (q_i, \nu_i, flag_i)$.*

Lemma 2. *For a NeTA-F $\mathcal{N}$, a DTPDA-F $\mathcal{E}(\mathcal{N})$, and configurations κ, κ' of $\mathcal{N}$,*

(Preservation) *if $\kappa \longrightarrow_{\mathcal{N}} \kappa'$, then $[\![\kappa]\!] \hookrightarrow^*_{\mathcal{E}(\mathcal{N})} [\![\kappa']\!]$, and*
(Reflection) *if $[\![\kappa]\!] \hookrightarrow^*_{\mathcal{N}} \varrho$, there exists κ' with $\varrho \hookrightarrow^*_{\mathcal{E}(\mathcal{N})} [\![\kappa']\!]$ and $\kappa \longrightarrow^*_{\mathcal{N}} \kappa'$.*

By this encoding, we have our main result from Theorem 1.

Theorem 3. *The reachability of a NeTA-F $(T, \mathcal{A}_0, C, \Delta)$ is decidable, if $|C| = 1$.*

5 Conclusion

This paper extends nested timed automata (NeTAs) to NeTA-Fs with *frozen local clocks*. A NeTA(-F) has a stack whose alphabet consists of timed automata. By the frozen clocks combined with value passing between clocks, past local clock values are recorded. The reachability of NeTA-F with 2 global clocks was shown to be undecidable by simulating the Minsky machine. However, with a single global clock, the reachability was shown to be *decidable*, by encoding NeTA-F to a snapshot PDS, which is a WSPDS with a well-formed constraint [9].

Acknowledgements This work is supported by the NSFC-JSPS bilateral joint research project (61511140100), NSFC(61472240, 91318301, 61261130589), and JSPS KAKENHI Grant-in-Aid for Scientific Research(B) (15H02684, 25280023) and Challenging Exploratory Research (26540026).

References

1. Alur, R., Dill, D.L.: A Theory of Timed Automata. Theoretical Computer Science **126** (1994) 183–235
2. Trivedi, A., Wojtczak, D.: Recursive Timed Automata. ATVA'10. LNCS 6252, Springer-Verlag (2010) 306–324
3. Benerecetti, M., Minopoli, S., Peron, A.: Analysis of Timed Recursive State Machines. TIME'10, IEEE (2010) 61–68
4. Li, G., Cai, X., Ogawa, M., Yuen, S.: Nested Timed Automata. FORMATS'13. LNCS 8053, Springer-Verlag (2013) 168–182
5. Cassez, F., Larsen, K.: The Impressive Power of Stopwatches. CONCUR'00. LNCS 1877, Springer-Verlag (2000) 138–152
6. Berard, B., Haddad, S., Sassolas, M.: Real Time Properties for Interrupt Timed Automata. TIME'10, IEEE (2010) 69–76
7. Krishna, S. N., Manasa L., Trivedi, A.: What's Decidable about Recursive Hybrid Automata? HSCC'15, ACM (2015) 31–40
8. Abdulla, P.A., Atig, M.F., Stenman, J.: Dense-Timed Pushdown Automata. LICS'12, IEEE (2012) 35–44
9. Cai, X., Ogawa, M.: Well-Structured Pushdown Systems: Case of Dense Timed Pushdown Automata. FLOPS'14. LNCS 8475, Springer-Verlag (2014) 336–352
10. Cai, X., Ogawa, M.: Well-Structured Pushdown Systems. In: Proceedings of CONCUR'13. LNCS 8052, Springer-Verlag (2013)
11. Leroux, J., Praveen, M., Sutre, G.: Hyper-Ackermannian Bounds for Pushdown Vector Addition Systems. CSL-LICS'14. IEEE (2014) 63:1–63:10
12. Henzinger, T.A., Nicollin, X., Sifakis, J., Yovine, S.: Symbolic Model Checking for Real-Time Systems. Information and Computation **111** (1994) 193–244
13. Ouaknine, J., Worrell, J.: On the Language Inclusion Problem for Timed Automata: Closing a Decidability Gap. LICS'04. IEEE (2004) 54–63
14. Abdulla, P., Jonsson, B.: Verifying Networks of Timed Processes. TACAS'98. LNCS 1384, Springer-Verlag (1998) 298–312
15. Abdulla, P., Jonsson, B.: Model Checking of Systems with Many Identical Time Processes. Theoretical Computer Science **290** (2003) 241–264
16. Minsky, M.L.: Computation: Finite and Infinite Machines. Prentice-Hall (1967)
17. Henzinger, T., Kopke, P., Puri, A., Varaiya, P.: What's Decidable about Hybrid Automata? Journal of Computer and System Sciences **57** (1998) 94–124