# The Essence of Region Abstraction for Timed Pushdown Automata

Yuya Uezato and Yasuhiko Minamide

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

**Abstract.** Abdulla et al. introduced a highly infinite model of computation *timed pushdown automata* and showed the surprising result that the reachability problem is decidable. While the idea of their *region abstraction* is intuitive, they needed the elaborated proofs by lacking the good correspondence between concrete and abstract domain.
By our thorough analysis of their construction, we capture its essence and propose *synchronized recursive timed automata*, which have new and unusual transitions compared to existing models. Even if our model is obtained by investigating their region abstraction, we significantly simplify the decidability proof of the reachability problem by considering backward simulation.

## 1 Introduction

Abdulla et al. introduced timed pushdown automata (TPDA), which extend both the theories of pushdown systems and timed automata [1]. TPDA are highly infinite from the two dimensions: the unboundedness of the stack, and the unboundedness and denseness of clock values (reals). From this infiniteness, their result of showing the decidability of the fundamental decision problem "reachability problem" of TPDA is quite surprising.

Their construction is based on the *region abstraction* technique known in the theory of timed automata [3]. It is nonstandard and intriguing to accommodate pushdown extensions of timed automata by: (i) lazy time elapsing technique, which is used to simulate TPDA with pushdown systems, (ii) a region keeps the fractional parts of clocks beyond the bound, (iii) the special clock, reference clock, which plays some important roles. On the other hand, the details of their construction are involved and their proofs are elaborated. Hence the considering further extensions is difficult.

By our thorough analysis of their construction, we extract the essence of the construction and propose a new model *synchronized recursive timed automata (SRTA)*. TPDA have two kinds of clocks, global clocks and local clocks. However, SRTA have only one kind of clocks that can behave as both global and local clocks, and this yields the uniform treatment of clocks and our simplified proofs. Our model has unusual transitions compared to the timed pushdown automata of Abdulla et al. and the recursive timed automata introduced by Trivedi and Wojtczak in [10], and Benerecetti et al. in [4]. The transitions in our model can

be seen as more primitive, hence we can simulate the transitions in their models by using our transitions.

We significantly simplify the decidability proof of the reachability problem by considering backward simulation. The result comes from (i) clarifying the essence of the construction of Abdulla et al. and (ii) dividing our proofs into the two steps. We first translate SRTA to an intermediate model by adopting their lazy time elapsing technique *without* applying the region abstraction, and then apply the region abstraction to obtain a finite model from the intermediate one. The proof of simulation properties of the second step is derived by combining the fundamental properties of regions. We obtain a simpler decidability proof while our formalization and construction are based on the Abdulla's work. Because of the lack of space, we omit all proofs of lemmas and theorems. They can be found in the longer version[1] of this paper.

The recursive timed automata (RTA) introduced in [10, 4] are related to SRTA. During of transitions of RTA, the time elapses for the clocks on the stack top only. The combination of this local-time elapse operation with diagonal constraints is very powerful. Indeed, they showed that the class of RTA is Turing-complete in general and the reachability problem of RTA is undecidable.

Cai and Ogawa [7] gave an alternative decidability proof of TPDA by applying the theory of well-structured transition systems [2, 8].

**Basic Terminology.** $\mathbb{Q}^+$ is the set of positive rationals, $\mathbb{Q}^+ \triangleq \{r \in \mathbb{Q} : 0 \leq r\}$. For every rational $r \in \mathbb{Q}^+$, $\lfloor r \rfloor$ and $\{r\}$ denote the integral part and the fractional part of $r$, respectively. For example, $\lfloor 1.5 \rfloor = 1$ and $\{1.5\} = 0.5$. Let $\mathsf{M} \in \mathbb{N}$ be a natural number. The subsets bounded by $\mathsf{M}$, $\mathbb{N}_{\mathsf{M}}$ and $\mathbb{Q}^+_{\mathsf{M}}$, are defined as $\mathbb{N}_{\mathsf{M}} \triangleq \{n \in \mathbb{N} : n \leq \mathsf{M}\}$, $\mathbb{Q}^+_{\mathsf{M}} \triangleq \{r \in \mathbb{Q}^+ : r \leq \mathsf{M}\}$. $A \uplus B$ is used for the union of two disjoint sets $A$ and $B$. Capital letters $X, Y, \ldots$ are used to denote a finite set of clocks.

**Concrete Valuations.** We write $\mathcal{V}_X$ for the set of concrete valuations over $X$: $\mathcal{V}_X \triangleq X \to \mathbb{Q}^+$ and omit the subscript when $X$ is clear from the context. Greek characters $\nu, \mu, \lambda, \ldots$ are used to denote concrete valuations.

Let $\nu \in \mathcal{V}_X$ be a concrete valuation over clock set $X$. For $x, y \in X$ and $r \in \mathbb{Q}^+$, we write $\nu[x := r]$ and $\nu[x := y]$ for the valuations defined by:

$$\nu[x := r](y) \triangleq \begin{cases} r & \text{if } y = x \\ \nu(y) & \text{otherwise,} \end{cases} \qquad \nu[x := y] \triangleq \nu[x := \nu(y)].$$

In order to move the clocks of $\nu$ forward by $\delta \in \mathbb{Q}^+$, we use $(\nu + \delta)(x) \triangleq \nu(x) + \delta$ and define the partial-ordering $\nu \leq \nu'$ by $\exists \delta \in \mathbb{Q}^+ . \nu' = \nu + \delta$.

## 2 Synchronized Recursive Timed Automata

We introduce a model of computation, *synchronized recursive timed automata*, which captures the essence of the region abstraction considered by Abdulla et al [1]. We first define $\mathsf{M}$-bounded constraints and collapsed valuations.

[1] `http://score.cs.tsukuba.ac.jp/~uezato/icalp2015-long.pdf`

**M-Bounded Constraints.** Let $X$ be a clock set and $\mathsf{M} \in \mathbb{N}$ be a natural number. The M-bounded constraints $\mathcal{C}_{X,\mathsf{M}}$ are obtained by the following grammar:

$$\varphi ::= \lfloor x \rfloor \bowtie k \mid \{x\} \bowtie \{y\} \mid \{x\} = 0 \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi$$
$$\text{where } \bowtie \in \{<, =, \neq, >\},\ x, y \in X, \text{ and } k \in \mathbb{N}_{\mathsf{M}}.$$

The constraints are the logical characterization of the region considered by Abdulla et al.

**Definition 1 (M-Bounded Collapsed Valuations).** The set of M-bounded collapsed valuations over $X$ is $\Lambda_X \triangleq X \to (\mathbb{N}^{\infty}_{\mathsf{M}} \times \mathbb{Q}^{+}_{[0,1)})$ where $\mathbb{N}^{\infty}_{\mathsf{M}} \triangleq \mathbb{N}_{\mathsf{M}} \cup \{\infty\}$ and $\mathbb{Q}^{+}_{[0,1)} \triangleq \mathbb{Q}^{+} \cap [0,1)$. Upright greek letters $\nu, \mu, \lambda, \ldots$ are used to denote collapsed valuations. We use term *valuations* to denote collapsed valuations rather than concrete valuations. We omit $X$ and write $\Lambda$ if $X$ is clear from the context.

Abusing the notation, we use $v.r$ to denote $(v, r) \in \mathbb{N}^{\infty}_{\mathsf{M}} \times \mathbb{Q}^{+}_{[0,1)}$. Moreover $\lfloor v.r \rfloor$ and $\{v.r\}$ are used to denote $v$ and $r$, respectively. ■

Let $r \in \mathbb{Q}^{+}$ and $\nu \in \mathcal{V}_X$. The collapsed value $\widehat{r}$ and valuation $\widehat{\nu}$ are obtained by:
– If $r \leq \mathsf{M}$, then $\widehat{r} \triangleq (\lfloor r \rfloor, \{r\})$. Otherwise, $\widehat{r} \triangleq (\infty, \{r\})$.
– By using this collapse operation, $\widehat{\nu}$ is defined by $\widehat{\nu}(x) \triangleq \widehat{r}$ where $r = \nu(x)$.

The fractional part is kept for the constraints $\{x\} = 0$ and $\{x\} \bowtie \{y\}$. Clearly, for any valuation $\nu$, there exists a concrete valuation $\nu$ such that $\nu = \widehat{\nu}$.

**Satisfiability $\nu \models \varphi$.** Let $\nu$ be a valuation over $X$ and $\varphi \in \mathcal{C}_X$. We define the satisfiability $\nu \models \varphi$ through the atomic cases: (i) $\nu \models \lfloor x \rfloor \bowtie k$ if $\lfloor \nu(x) \rfloor \bowtie k$, (ii) $\nu \models \{x\} \bowtie \{y\}$ if $\{\nu(x)\} \bowtie \{\nu(y)\}$, (iii) $\nu \models \{x\} = 0$ if $\{\nu(x)\} = 0$.

**Time passage and quasi-ordering.** We define the time-passage $\nu + \delta$ for a valuation $\nu$ and $\delta \in \mathbb{Q}^{+}$ by $\nu + \delta \triangleq \widehat{\nu + \delta}$ where $\nu$ is a witness such that $\nu = \widehat{\nu}$. $\nu + \delta$ is well-defined because the result does not depend on the choice of a witness. We define the *quasi*-ordering $\nu \preccurlyeq \nu'$ if there exists $\delta \in \mathbb{Q}^{+}$ such that $\nu' = \nu + \delta$. For $\nu = \{x \mapsto \infty.0\}$ and $\mu = \{x \mapsto \infty.5\}$, $\nu \preccurlyeq \mu$ and $\nu \succcurlyeq \mu$. However, $\nu \neq \mu$.

**Updating $\nu[x := (v, r)]$ and $\nu[x := y]$.** The updating $\nu[x := (v, r)]$ and $\nu[x := y]$ are defined in the same way as concrete valuations.

**Restriction $\nu|Y$.** For a valuation $\nu \in \Lambda_X$ and $Y \subseteq X$, we use $\nu|Y \in \Lambda_Y$ to denote the restricted valuation of $\nu$ whose domain is $Y$. We write $\nu \oplus \mu$ instead of $\nu \cup \mu$ for valuations $\nu \in \Lambda_{Y_1}$ and $\mu \in \Lambda_{Y_2}$ such that $\nu|Y_1 \cap Y_2 = \mu|Y_1 \cap Y_2$.

**Definition 2 (Synchronized Recursive Timed Automata).** A synchronized recursive timed automata (SRTA) is a structure $(Q, \Gamma, \mathsf{M}, \Delta)$ where $Q$ is a finite set of control locations, $\Gamma$ is a finite set of clocks, $\mathsf{M}$ is a natural number, and $\Delta \subseteq Q \times Act \times Q$ is a finite set of transition rules. The set $Act$ is given by:

$$Act ::= \mathbf{time} \mid \mathbf{push} \mid \mathbf{return} \mid \mathbf{dig}(x, y) \mid \mathbf{upd}(X, \varphi) \quad \text{where } x, y \in \Gamma, X \subseteq \Gamma, \varphi \in \mathcal{C}_{\Gamma,\mathsf{M}}.$$

**Remark:** We assume the special clock $\complement$ belongs to $\Gamma$. It is called *reference clock* in the Abdulla's construction and it plays some important roles. ■

We present the essence of Abdulla's construction by introducing the collapsed semantics of SRTA.

**Definition 3 (Collapsed Semantics $\mathcal{S}_{\mathbf{coll}}$).** A configuration is a pair $\langle q, \omega \rangle$ of a control location $q$ and a stack $\omega$ whose element is a valuation $\nu \in \Lambda_\Gamma$. The set of configurations $\mathit{Conf}(\mathcal{S}_{\mathbf{coll}})$ is $Q \times \Lambda^*$.

For $\tau \in \mathit{Act}$, we define the action $\omega \xrightarrow{\tau} \omega'$ with $\omega, \omega' \in \Lambda^*$ by case analysis on $\tau$ as follows (we omit $\tau$ from $\omega \xrightarrow{\tau} \omega'$ and write $\omega \to \omega'$):

**Case time:** $\omega\, \nu \to (\omega\, \nu){+}\delta$ where $\delta \in \mathbb{Q}^+$ and $(\nu_1 \ldots \nu_n){+}\delta \triangleq (\nu_1{+}\delta) \ldots (\nu_n{+}\delta)$.
**Case push:** $\omega\, \nu \to \omega\, \nu'\, 0_\Lambda$ where $0_\Lambda(x) \triangleq 0$ and $\nu' = \nu[\complement := 0]$.
**Case return:** $\omega\, \nu\, \mu \to \omega\, \mu$.
**Case dig**$(x, y)$**:** $\omega\, \nu\, \mu \to \omega\, \nu\, (\mu[x := \nu(y)])$.
**Case upd**$(X, \varphi)$**:** $\omega\, \nu \to \omega\, \nu'$ where $\nu' \in \nu[X \leftarrow \varphi]$. For $X = \{x_1, x_2, \ldots, x_n\}$,

$$\nu[X \leftarrow \varphi] \triangleq \left\{ \nu' : r_i \in \mathbb{Q}^+, \nu' = \nu[x_1 := r_1, x_2 := r_2, \ldots, x_n := r_n], \nu' \models \varphi \right\}.$$

The transition relation $\xrightarrow{\tau} \subseteq \mathit{Conf}(\mathcal{S}_{\mathbf{coll}}) \times \mathit{Conf}(\mathcal{S}_{\mathbf{coll}})$ is defined as follows:

$$\langle q_1, \omega_1 \rangle \xrightarrow{\tau} \langle q_2, \omega_2 \rangle \qquad \text{if } \langle q_1, \tau, q_2 \rangle \in \Delta \text{ and } \omega_1 \xrightarrow{\tau} \omega_2. \qquad \blacksquare$$

By **time**, the valuations in the stack evolve synchronously. The **push** transition resets the reference clock and pushes $0_\Lambda$ onto the stack. This reset is crucial in Abdulla's construction. The **return** transition removes the second valuation $\nu$, and then the top valuation $\mu$ stays on the top. The **dig**$(x, y)$ transition is peculiar and we pass the value of $y$ in the second valuation $\nu$ into $x$ in the top valuation $\mu$. The **upd**$(X, \varphi)$ transition updates and checks the top valuation $\nu$ at the same time. The top valuation $\nu$ can be checked without modification by **upd**$(\emptyset, \varphi)$.

**Usage of the dig transition.** By combining the **dig** and **return** transitions, the usual **pop** transition is implemented as follows:

$$\omega\, \nu\, \mu \xrightarrow{\mathsf{pop}} \omega\, \nu \text{ is simulated by } \omega\, \nu\, \mu \xrightarrow{\mathsf{dig}(x_1, x_1)} \cdots \xrightarrow{\mathsf{dig}(x_n, x_n)} \omega\, \nu\, \nu \xrightarrow{\mathsf{return}} \omega\, \nu,$$

where $\Gamma = \{x_1, \ldots, x_n\}$. With the same idea, we can simulate the pass-by-value and pass-by-reference appearing in recursive timed automata [10, 4].

**Encoding Stack Symbols.** Let us see the example with stack symbols $\{a, b\}$:

$$\omega\, \langle a, \nu \rangle \xrightarrow{\mathsf{push}(b)} \omega\, \langle a, \nu \rangle\, \langle b, 0_\Lambda \rangle \xrightarrow{\mathsf{test}(b)} \omega\, \langle a, \nu \rangle\, \langle b, 0_\Lambda \rangle.$$

This pushes the stack symbol $b$ and tests it against $b$. This is encoded by:

$$\omega'\, \nu' \xrightarrow{\mathsf{push}} \omega'\, \nu'\, 0_\Lambda \xrightarrow{\mathsf{upd}(\{\sharp, \sharp_a, \sharp_b\}, \psi_b)} \omega'\, \nu'\, \mu \xrightarrow{\mathsf{upd}(\emptyset, \{\sharp\} = \{\sharp_b\})} \omega'\, \nu'\, \mu,$$

where $\psi_b \triangleq \{\sharp\} \neq \{\sharp_a\} \wedge \{\sharp\} = \{\sharp_b\}$ and $b$ is encoded in the fractional parts.

**Reachability Problem of SRTA.** Let $q_{\mathrm{init}}, q_{\mathrm{final}} \in Q$ be control locations. The *reachability problem from* $q_{\mathrm{init}}$ *to* $q_{\mathrm{final}}$ is to decide whether there is a computation from $\langle q_{\mathrm{init}}, 0_\Lambda \rangle$ to $\langle q_{\mathrm{final}}, \omega \rangle$ for some stack $\omega \in \Lambda^*$.

### 2.1 Comparison Study: Encoding Timed Pushdown Automata

We review the definition of TPDA introduced by Abdulla et al. [1], and see how a TPDA is simulated as a SRTA. It should be noted that the following definition is slightly revised from the original, since in their model a clock constraint is of the form $c \in_? I$ that checks whether the value of $c$ belongs to an interval $I$.

A TPDA is a structure $(Q, \Sigma, \bot, \mathcal{X}, \mathsf{M}, \Delta)$ where $\Sigma$ is a finite set of stack symbols, $\bot \in \Sigma$ is the identified stack symbol, $\mathcal{X}$ is a finite set of global clocks, and $\Delta \subseteq Q \times Disc \times Q$ is a finite set of transition rules. The set $Disc$ is given by:

$$Disc ::= \mathsf{push}(\sigma, \phi) \mid \mathsf{pop}(\sigma, \phi) \mid \mathsf{upd}(X, \varphi) \quad \text{where } \sigma \in \Sigma, \phi \in \mathcal{C}_\sigma, X \subseteq \mathcal{X}, \varphi \in \mathcal{C}_\mathcal{X}.$$

A stack element is a pair $\langle \sigma, r \rangle$ of a stack symbol $\sigma \in \Sigma$ and a rational $r \in \mathbb{Q}^+$, and a configuration is a triple $\langle q, \nu, w \rangle$ of a location $q \in Q$, a concrete valuation of global clocks $\nu \in \mathcal{V}_\mathcal{X}$, and a stack $w \in (\Sigma \times \mathbb{Q}^+)^*$.

For $\langle p, \tau, q \rangle \in \Delta$, we define transitions by case analysis on $\tau$ as follows:

**Case** $\mathsf{push}(\sigma, \phi)$**:** $\langle p, \nu, w \rangle \rightsquigarrow \langle q, \nu, w\,\langle \sigma, r \rangle \rangle$ where $r \in \mathbb{Q}^+$ with $\{\sigma \mapsto r\} \models \phi$.
**Case** $\mathsf{pop}(\sigma, \phi)$**:** $\langle p, \nu, w\,\langle \sigma, r \rangle \rangle \rightsquigarrow \langle q, \nu, w \rangle$ if $\{\sigma \mapsto r\} \models \phi$.
**Case** $\mathsf{upd}(X, \varphi)$**:** $\langle p, \nu, w \rangle \rightsquigarrow \langle q, \nu', w \rangle$ where $\nu' \in \nu[X \leftarrow \varphi]$.

In the above, we use the satisfiability $\nu \models \varphi$ and $\nu[X \leftarrow \varphi]$ for concrete valuations. They are defined in a similar way to collapsed valuations. The $\mathsf{push}(\sigma, \phi)$ transition pushes $\langle \sigma, r \rangle$, which satisfies $\phi$. The $\mathsf{pop}(\sigma, \phi)$ transition pops the top $\langle \sigma, r \rangle$ if $r$ satisfies $\phi$.

Their formulation does not permit the explicit **time**-transition but the corresponding transition $\overset{time}{\rightsquigarrow}$ invokes *spontaneously*: $\langle p, \nu, w \rangle \overset{time}{\rightsquigarrow} \langle p, \nu + \delta, w + \delta \rangle$ for some $\delta \in \mathbb{Q}^+$. The transition relation $\rightsquigarrow$ is defined as follows:

$$\langle p, \nu, w \rangle \rightsquigarrow \langle q, \nu', w' \rangle \quad \text{if } \exists \langle p, \tau, q \rangle \in \Delta. \langle p, \nu, w \rangle \overset{\tau}{\rightsquigarrow} \langle q, \nu', w' \rangle \text{ or } \langle p, \nu, w \rangle \overset{time}{\rightsquigarrow} \langle q, \nu', w' \rangle.$$

**Reachability Problem of TPDA.** Let $q_{\text{init}}, q_{\text{final}} \in Q$ be control locations. The *reachability problem from* $q_{\text{init}}$ *to* $q_{\text{final}}$ is to decide whether there is a computation from $\langle q_{\text{init}}, 0_\mathcal{V}, \langle \bot, 0.0 \rangle \rangle \rightsquigarrow^* \langle q_{\text{final}}, \nu, w \rangle$ for some valuation $\nu \in \mathcal{V}_\mathcal{X}$ and stack $w \in (\Sigma \times \mathbb{Q}^+)^*$.

We use a SRTA $(Q', \mathcal{X} \cup \{\sharp, \complement\}, \mathsf{M}, \Delta')$ with the stack symbols $\Sigma$ and define the correspondence between configurations of RTA and the collapsed semantics:

– $\langle q, \nu, w\,\langle \sigma, r \rangle \rangle \sim \langle q, \boldsymbol{\omega}\,\langle \sigma, \boldsymbol{\nu} \rangle \rangle$ if $\widehat{\nu} = \boldsymbol{\nu}|\mathcal{X}$ and $w\langle \sigma, r \rangle \approx \boldsymbol{\omega}\langle \sigma, \boldsymbol{\nu} \rangle$.
– $\langle \sigma, r \rangle \approx \langle \sigma, \boldsymbol{\nu} \rangle$ if $\widehat{r} = \boldsymbol{\nu}(\sharp)$. $\quad w\langle \sigma, r \rangle \approx \boldsymbol{\omega}\langle \sigma, \boldsymbol{\nu} \rangle$ if $\langle \sigma, r \rangle \approx \langle \sigma, \boldsymbol{\nu} \rangle$ and $w \approx \boldsymbol{\omega}$.

The valuation $\nu$ of the global clocks is kept in the top valuation $\boldsymbol{\nu}$ in SRTA. We can encode TPDA into the corresponding SRTA based on the above correspondence so that the following are equivalent:

– There exists a computation $q_{\text{init}}$ from $q_{\text{final}}$ in a TPDA.
– There exists a computation $q_{\text{init}}$ from $q_{\text{final}}$ in the corresponding SRTA.

It is required to assume that $\langle q, \nu, w \rangle \rightsquigarrow^* \langle q', \nu', \epsilon \rangle$ never occur. Since TPDA can perform a transition from a configuration with *the empty stack*, but SRTA cannot. Without loss of generality, we build the corresponding TPDA which satisfies the condition.

### 2.2 Comparison Study: Recursive Timed Automata [10, 4]

We review the recursive timed automata (RTA), which are independently introduced by Trivedi and Wojtczak in [10] and Benerecetti et al. in [4]. A RTA can be seen as a structure $(Q, X, \Delta)$ and a configuration is a pair $\langle q, w \rangle$ of a location $q$ and a stack $w \in \mathcal{V}_X^*$. There are two significant differences between RTA and SRTA:

– RTA has the *local* time elapse transition $\langle p, \mathbf{Ltime}, q\rangle \in \Delta$, which invokes the transition $\langle p, w\,\nu\rangle \xrightarrow{\mathbf{Ltime}} \langle q, w\,(\nu+\delta)\rangle$ with some $\delta \in \mathbb{Q}^+$.
– Two clocks are compared by diagonal constraints such as $x_1 = x_2$, $x_1 < x_2$, etc.

The combination of local time elapse transition and the diagonal constraints is very powerful. Indeed, the undecidability of the reachability problem of RTA was proved by showing that RTA can encode two-counter machines in [10, 4]. However, the expressiveness of only using local time elapse transition, or the combination of **time** and diagonal constraints has not been studied yet. This will be discussed in the conclusion section.

## 3 Lazy Semantics and What Good is the Reference Clock?

The denseness of the rationals makes SRTA an infinite model, and the entire modification of the stack in the **time** transition makes SRTA beyond the theory of pushdown systems. As a starting point in attacking these problems, we distill the entire modification into the lazy semantics $\mathcal{S}_{\mathbf{lazy}}$. By considering the lazy semantics, the key role of the reference clock is revealed.

We quickly review the lazy time elapsing technique introduced by Abdulla et al. [1]. This removes the entire stack modification in the **time** transition.

For a preparation, we define the quasi-ordering on pairs of valuations: $\langle \nu, \mu\rangle \preccurlyeq \langle \nu', \mu'\rangle$ if $\exists \delta \in \mathbb{Q}^+.\, \nu' = \nu + \delta$ and $\mu' = \mu + \delta$. We say that $\langle \nu, \mu_1\rangle$ and $\langle \mu_2, \lambda\rangle$ are compatible if $\mu_1 = \mu_2$.

Let us explain the key idea of their construction by considering the following:

$$\omega\,\nu\,\mu \xrightarrow{\mathbf{push}} \omega\,\nu\,\mu'\,0_\Lambda \to \cdots \to \omega\,\nu\,\mu'\,\lambda \xrightarrow{\mathbf{time}} \omega'\,\nu'\,\mu''\,\lambda' \xrightarrow{\mathbf{return}} \omega'\,\nu'\,\lambda'.$$

This is simulated as follows by pairing the valuations:

$$\boldsymbol{w}\langle \nu, \mu\rangle \xrightarrow{\mathbf{push}} \boldsymbol{w}\langle \nu, \mu'\rangle\langle \mu', 0_\Lambda\rangle \to \cdots \to \boldsymbol{w}\langle \nu, \mu'\rangle\langle \mu', \lambda\rangle \xrightarrow{\mathbf{time}} \boldsymbol{w}\langle \nu, \mu'\rangle\langle \mu'', \lambda'\rangle \xrightarrow{\mathbf{return}} \boldsymbol{w}\langle \nu', \lambda'\rangle.$$

In the transition $\boldsymbol{w}\langle \nu, \mu'\rangle\langle \mu'', \lambda'\rangle \xrightarrow{\mathbf{return}} \boldsymbol{w}\langle \nu', \lambda'\rangle$, we first increase $\langle \nu, \mu'\rangle$ into $\langle \nu', \mu''\rangle$ with $\langle \nu, \mu'\rangle \preccurlyeq \langle \nu', \mu''\rangle$ to be compatible with $\langle \mu'', \lambda'\rangle$, and then we remove $\mu''$. This lazy time elapsing technique ensures the top pair $\langle \mu'', \lambda'\rangle$ in the simulating side is same as the top valuation $\lambda'$ and the second valuation $\mu''$ in the collapsed semantics. This idea derives the lazy semantics.

**Definition 4 (Lazy Semantics $\mathcal{S}_{\mathbf{lazy}}$).** A stack element is a pair $\langle \nu, \mu\rangle$ of valuations $\nu, \mu \in \Lambda_\Gamma$. A configuration is a pair $\langle q, \boldsymbol{w}\rangle$ of a location $q \in Q$ and a stack $\boldsymbol{w} \in (\Lambda_\Gamma \times \Lambda_\Gamma)^*$. The set of configurations $\mathit{Conf}(\mathcal{S}_{\mathbf{lazy}})$ is $Q \times (\Lambda \times \Lambda)^*$.

The action $\boldsymbol{w} \xrightarrow{\tau} \boldsymbol{w'}$ between stacks is defined by case analysis on $\tau \in \mathit{Act}$:

**Case time:** $\boldsymbol{w}\,\langle \nu, \mu\rangle \to \boldsymbol{w}\,\langle \nu', \mu'\rangle$ where $\langle \nu, \mu\rangle \preccurlyeq \langle \nu', \mu'\rangle$.
**Case push:** $\boldsymbol{w}\langle \nu, \mu\rangle \to \boldsymbol{w}\langle \nu, \mu'\rangle\langle \mu', 0_\Lambda\rangle$ where $\mu' = \mu[\complement := 0]$.
**Case return:** $\boldsymbol{w}\,\langle \nu, \mu\rangle\,\langle \mu', \lambda\rangle \to \boldsymbol{w}\,\langle \nu', \lambda\rangle$ where $\langle \nu, \mu\rangle \preccurlyeq \langle \nu', \mu'\rangle$.
**Case dig**$(x, y)$**:** $\boldsymbol{w}\langle \nu, \mu\rangle\langle \mu', \lambda\rangle \to \boldsymbol{w}\langle \nu, \mu\rangle\langle \mu', \lambda[x := \mu'(y)]\rangle$.
**Case upd**$(X, \varphi)$**:** $\boldsymbol{w}\,\langle \nu, \mu\rangle \to \boldsymbol{w}\,\langle \nu, \mu'\rangle$ where $\mu' \in \mu[X \leftarrow \varphi]$. ∎

We define the well-formedness of a stack, $\mathsf{WF}(\boldsymbol{w})$, inductively by

$$\mathsf{WF}(\langle \nu, \mu\rangle), \qquad \mathsf{WF}(\boldsymbol{w}\,\langle \nu, \mu\rangle\,\langle \mu', \lambda\rangle) \text{ if } \mu(\complement) = 0,\ \mu \preccurlyeq \mu', \text{ and } \mathsf{WF}(\boldsymbol{w}\,\langle \nu, \mu\rangle).$$

From the definition of the actions, it is clear that every transition preserves well-formedness. That is, if $\mathsf{WF}(\boldsymbol{w})$ and $\boldsymbol{w} \to \boldsymbol{w'}$, then $\mathsf{WF}(\boldsymbol{w'})$.

**The Correspondence between $\mathcal{S}_{\mathbf{coll}}$ and $\mathcal{S}_{\mathbf{lazy}}$.** Let $\omega$ and $\boldsymbol{w}$ be stacks of $\mathcal{S}_{\mathbf{coll}}$ and $\mathcal{S}_{\mathbf{lazy}}$ with $\mathsf{WF}(\boldsymbol{w})$. We introduce the correspondence relation $\omega \sim \boldsymbol{w}$.

First, we define the concretization function $\Phi$ to concretize a stack of lazy semantics. For a well-formed stack $\mathsf{WF}(\boldsymbol{w})$, the function $\Phi : (\Lambda \times \Lambda)^* \to \Lambda^*$ is defined inductively as follows:

$$\Phi(\langle \nu, \mu \rangle) \triangleq \mu, \qquad \Phi(\boldsymbol{w}\, \langle \nu, \mu \rangle\, \langle \mu', \lambda \rangle) \triangleq \Phi(\boldsymbol{w}\, \langle \nu', \mu' \rangle)\, \lambda,$$

where $\nu'$ is the valuation such that $\langle \nu, \mu \rangle \preccurlyeq \langle \nu', \mu' \rangle$.

*The role of the reference clock.* Note that such $\nu'$ is unique because $\mu(\mathbb{C}) = 0$ for $\mathsf{WF}(\boldsymbol{w}\langle \nu, \mu \rangle \langle \mu', \lambda \rangle)$. The following proposition ensures this uniqueness.

**Proposition 5.** If $\mu(\mathbb{C}) = 0$ and $\langle \nu, \mu \rangle \preccurlyeq \langle \nu_1, \mu' \rangle, \langle \nu_2, \mu' \rangle$, then $\nu_1 = \nu_2$.

If we lack the reference clock, then the following bad situation occurs with $\mathsf{M} = 2$:

$$\langle \{x \mapsto 1.0\}, \{y \mapsto 2.0\} \rangle \preccurlyeq \langle \{x \mapsto 2.0\}, \{y \mapsto \infty.0\} \rangle, \langle \{x \mapsto \infty.0\}, \{y \mapsto \infty.0\} \rangle.$$

The concretization function derives the forward simulation properties.

**Theorem 6.** Let $\omega$ and $\boldsymbol{w}$ be stacks of $\mathcal{S}_{\mathbf{coll}}$ and $\mathcal{S}_{\mathbf{lazy}}$, respectively.
If $\mathsf{WF}(\boldsymbol{w})$ and $\omega = \Phi(\boldsymbol{w})$, then the following hold:

1. If $\omega \xrightarrow{\tau} \omega'$, then there exists $\boldsymbol{w'}$ such that $\boldsymbol{w} \xrightarrow{\tau} \boldsymbol{w'}$ and $\omega' = \Phi(\boldsymbol{w'})$.
2. If $\boldsymbol{w} \xrightarrow{\tau} \boldsymbol{w'}$, then there exists $\omega'$ such that $\omega \xrightarrow{\tau} \omega'$ and $\omega' = \Phi(\boldsymbol{w'})$.

This result states that we can use the lazy semantics instead of the collapsed semantics to solve the reachability problem. However, the lazy semantics is infinite from the denseness of the fractional parts of clocks. In the next section, we abstract the lazy semantics into a finite one by applying the region abstraction technique.

## 4 Digital Valuation: Revisiting Abdulla's Region

We define a finite abstraction of valuations $\Lambda$ to abstract the lazy semantics. To ensure the finiteness of our abstraction, we adopt the same construction technique, region abstraction, used by Abdulla et al. [1]. It abstracts the concrete fractional values of clocks away and only keeps the ordering information between the clocks.

First, we define the digital valuations as the abstract domain of valuations. The structure of digital valuations is basically the same as the region considered in [1]. Since the regions considered by Abdulla et al. contain additional information, we use the term *digital* valuations to distinguish theirs.

**Definition 7 (Digital Valuations).** Let $X$ be a clock set and $D$ be a sequence of clock sets $c_0\, c_1 \ldots c_n$ where $c_i \subseteq X \times \mathbb{N}^{\infty}_{\mathsf{M}}$. First, two useful notations are defined:

$$\begin{array}{ll} x \in c_i & \text{if } \exists v \in \mathbb{N}^{\infty}_{\mathsf{M}}.\, (x, v) \in c_i, \\ (x, v) \in D & \text{if } \exists i.\, (x, v) \in c_i. \end{array}$$

$D$ is a digital valuation if it satisfies the following:

- Every clock appears: if $x \in X$, then $x \in c_i$ with $0 \leq i \leq n$.
- Every clock appears once: if $(x, v) \in c_i$ and $(x, v') \in c_j$, then $i = j$ and $v = v'$.

– $D$ is almost normalized: $c_i \neq \emptyset$ for $1 \leq i \leq n$. Note $c_0$ may be $\emptyset$.
– $\mathsf{M}$ appears at the head position only: if $(x, \mathsf{M}) \in D$, then $(x, \mathsf{M}) \in c_0$.

We use $\mathcal{D}_X$ to denote the set of digital valuations over $X$. ∎

Intuitively, a valuation $\nu = \{x \mapsto 0.8, y \mapsto 1.2, z \mapsto \infty.5\}$ is abstracted into the digital valuation $D = \emptyset\,\{(y,1)\}\,\{(z,\infty)\}\,\{(x,0)\}$. Then, we formalize this intuition and define the abstraction function $\alpha$ from a valuation $\nu$ to the corresponding digital valuation $D$.

**Abstraction.** Let $\nu$ be a valuation and $D = c_0\, c_1 \ldots c_n$ be a digital valuation. Then, the realization relation $\nu \models D$ holds if the following are satisfied:
(a) $(x, \lfloor \nu(x) \rfloor) \in D$, (b) $\{\nu(x)\} = 0$ iff $x \in c_0$, (c) $\{\nu(x)\} \bowtie \{\nu(y)\}$ iff $x \in c_i$ and $y \in c_j$ with $i \bowtie j$.

**Proposition 8.** Valuations and digital valuations are closely related as follows.
– For a digital valuation $D$, there exists a valuation $\nu$ such that $\nu \models D$.
– For a valuation $\nu$, there exists the unique digital valuation $\alpha(\nu)$ with $\nu \models \alpha(\nu)$.

### 4.1 Basic Operations of Digital Valuations

We define the basic operations of digital valuations that correspond to $\nu \models \varphi$, $\nu \preccurlyeq \nu'$, $\nu[x := y]$, $\nu[X \leftarrow \varphi]$, and $\nu|X$ for valuations.

Let $X$ be a clock set and $D = c_0 \ldots c_n$ be a digital valuation over $X$.

**Satisfiability $D \models \varphi$.** Let $x \in X$ be a clock and $\varphi \in \mathcal{C}_X$ be a constraint. We define $D \models \varphi$ as follows:

$$\begin{array}{ll} D \models \lfloor x \rfloor \bowtie k & \text{if } (x, k') \in D \text{ and } k' \bowtie k, \\ D \models \{x\} \bowtie \{y\} & \text{if } x \in c_i \text{ and } y \in c_j \text{ with } i \bowtie j, \\ D \models \{x\} = 0 & \text{if } x \in c_0. \end{array}$$

The inductive cases are defined naturally. Note that $c_0$ plays a distinguished role to define $\{x\} = 0$. From the definition, it is clear that $\nu \models \varphi$ iff $\alpha(\nu) \models \varphi$.

**Time Passage.** We define the time-passage relation $D \vdash D'$, which corresponds to $\nu \preccurlyeq \nu'$, as follows:

*Small elapse:* if $c_0 \neq \emptyset$, then $c_0\, c_1 \ldots c_n \vdash \emptyset\, c_0'\, c_1 \ldots c_n$ where $c_0'$ is defined by:

$$(x, \mathsf{M}) \in c_0 \Rightarrow (x, \infty) \in c_0', \qquad (x, v) \in c_0 \Rightarrow (x, v) \in c_0' \ (v \in \mathbb{N}_{\mathsf{M}}^{\infty} \wedge v \neq \mathsf{M}).$$

*Carry:* $\emptyset\, c_1 \ldots c_n\, c_{n+1} \vdash c_0'\, c_1 \ldots c_n$ where $c_0'$ is defined as follows:

$$(x, \infty) \in c_{n+1} \Rightarrow (x, \infty) \in c_0', \qquad (x, v) \in c_{n+1} \wedge v < \mathsf{M} \Rightarrow (x, v+1) \in c_0'.$$

We use $D \vdash^* D'$ to denote the reflexive transitive closure of $\vdash$.

**Proposition 9.** The correspondences between the relation $\vdash$ and $\preccurlyeq$ are:

1. If $\nu \models D$ and $D \vdash D'$, then there exists $\nu'$ such that $\nu' \models D'$ and $\nu \preccurlyeq \nu'$.
2. If $\nu \models D$ and $\nu \preccurlyeq \nu'$, then there exists $D'$ such that $\nu' \models D'$ and $D \vdash^* D'$.
3. If $\nu' \models D'$ and $D \vdash D'$, then there exists $\nu$ such that $\nu \models D$ and $\nu \preccurlyeq \nu'$.
4. If $\nu' \models D'$ and $\nu \preccurlyeq \nu'$, then there exists $D$ such that $\nu \models D$ and $D \vdash^* D'$.

The forward correspondences 1 and 2 are standard for regions considered in the theory of timed automata [3]. However, the backward correspondences 3 and 4 are missed in *concrete* valuations. Let us explain the example with $\mathsf{M} = 1$, $D = \{(x,0),(y,1)\} \vdash D' = \emptyset\,\{(x,0),(y,\infty)\}$.

Then $\nu' = \{x \mapsto 0.1, y \mapsto 2.1\} \models D'$, but there is no $\nu$ such that $\nu \leq \nu'$ and $\nu \models D$. For the collapsed valuation $\widehat{\nu}' = \{x \mapsto 0.1, y \mapsto \infty.1\} \models D'$, we can find $\nu = \{x \mapsto 0.0, y \mapsto 1.0\}$ such that $\nu \preccurlyeq \widehat{\nu}'$ and $\nu \models D$.

The backward correspondences 3 and 4 are quite important in our proof of the main Theorem 14. Instead of forward simulation, Theorem 14 establishes a backward simulation which requires the above backward correspondences.

**Updating $D[x{:=}y]$ and $D[X \leftarrow \varphi]$.** A digital valuation $D[x := y] \in \mathcal{D}_X$ is obtained by copying the value of $y$ into $x$: $D[x := y] \triangleq \{\alpha(\nu') : \nu \models D, \nu' = \nu[x := y]\}$. Note that $D[x := y]$ is singleton and we use $D[x := y]$ to denote such the element.

A finite set of digital valuations $D[X \leftarrow \varphi]$ is obtained by updating $X$ to satisfy the constraint $\varphi$: $D[X \leftarrow \varphi] \triangleq \{\alpha(\nu') : \nu \models D, \nu' \in \nu[X \leftarrow \varphi]\}$. We can define $D[x := y]$ and $D[X \leftarrow \varphi]$ in a more concrete manner, but here we adopt the above one to save space.

**Restriction.** Let $Y \subseteq X$. The restriction $D|Y$ of $D$ to $Y$ is defined by $D|Y \triangleq \{\alpha(\nu|Y) : \nu \models D\}$. Then $D|Y$ is singleton and we use $D|Y$ to denote the element.

**Proposition 10.** The following are basic properties of the restriction.

- If $\nu \in \Lambda_X$, $Y \subseteq X$, $\nu \models D$, then $\nu|Y \models D|Y$.
- Let $D \in \mathcal{D}_X$, $D' \in \mathcal{D}_Y$, and $X \subseteq Y$.
  If $\nu \models D$ and $D = D'|X$, then there exists $\nu'$ such that $\nu = \nu'|X$ and $\nu' \models D'$.

These notions have been considered in the theory of timed automata, and thus they are not intrinsic operations for the pushdown extension of timed automata. By contrast, the notations below are introduced for the pushdown extension. Basically they are already considered by Abdulla et al., but we present the same thing in a simpler way.

**Renaming Clock.** Let $\rho : X \leftrightarrows Y$ be a renaming (bijection) from $X$ to $Y$. For a digital valuation $D \in \mathcal{D}_X$, we define the renamed digital valuation $\rho(D) \in \mathcal{D}_Y$:

$$\rho(c_0\, c_1 \ldots c_n) \triangleq \rho(c_0)\, \rho(c_1)\, \ldots\, \rho(c_n),$$

where $\rho(c_i) \triangleq \{(\rho(x), v) : (x, v) \in c_i\} \subseteq Y \times \mathbb{N}^{\infty}_{\mathsf{M}}$.

**Compatibility and Join.** Let $D_1 \in \mathcal{D}_X$ and $D_2 \in \mathcal{D}_Y$ be digital valuations.

$D_1$ and $D_2$ are compatible or joinable if $D_1|X \cap Y = D_2|X \cap Y$, and this is written by $D_1 \parallel D_2$. We define the join operation $D_1 \oplus D_2$ in order to obtain a large digital valuation from small digital valuations:

$$D_1 \oplus D_2 \triangleq \{D \in \mathcal{D}_{X \cup Y} : D_1 = D|X,\ D_2 = D|Y\}.$$

It is clear that $D_1 \parallel D_2$ iff $D_1 \oplus D_2$ is not empty.

**Proposition 11.** Let $D_1 \in \mathcal{D}_X$ and $D_2 \in \mathcal{D}_Y$ be digital valuations.

- If $\nu \models D_1$, $\mu \models D_2$, and $\nu \oplus \mu$ is defined, then $\alpha(\nu \oplus \mu) \in D_1 \oplus D_2$.
- If $D \in D_1 \oplus D_2$, then we can find $\nu_i$ such that $\nu_i \models D_i$ and $\nu_1 \oplus \nu_2 \models D$.

## 5 Digitized Semantics

In the lazy semantics, a stack element is a pair of valuations, $\langle \nu, \mu \rangle$. In order to abstract the pair into a *single* digital valuation $D$, we define a marker.

**Marked Clock.** We write $\dot{x}$ and $\underset{\cdot}{x}$ to denote the marked clocks of $x$. The marked clock set $\dot{X}$ is defined by $\dot{X} \triangleq \{\dot{x} : x \in X\}$, and $\underset{\cdot}{X}$ is also defined. For a valuation $\nu \in \Lambda_X$, we use $\dot{\nu} \in \Lambda_{\dot{X}}$ and $\underset{\cdot}{\nu} \in \Lambda_{\underset{\cdot}{X}}$ to denote the marked valuations. For a constraint $\varphi \in \mathcal{C}_X$, $\dot{\varphi} \in \mathcal{C}_{\dot{X}}$ is the constraint obtained by marking all clocks in $\varphi$.

A pair of valuations $\langle \nu, \mu \rangle$ is isomorphic to $\underset{\cdot}{\nu} \oplus \dot{\mu}$. By using this, $\langle \nu, \mu \rangle$ is abstracted into $\alpha(\underset{\cdot}{\nu} \oplus \dot{\mu})$ and we write $\langle \nu, \mu \rangle \models D$ if $\underset{\cdot}{\nu} \oplus \dot{\mu} \models D$.

For example, the following holds:

$$\langle \{x \mapsto 0.0, y \mapsto 1.5\}, \{x \mapsto \infty.2, y \mapsto 0.7\} \rangle \models \{(\underset{\cdot}{x}, 0)\}\{(\dot{x}, \infty)\}\{(\underset{\cdot}{y}, 1)\}\{(\dot{y}, 0)\}.$$

Next, we prepare the marker removing operation by using renaming.

**Removing Marker.** Let $D \in \mathcal{D}_{\Gamma \cup \dot{\Gamma}}$ be a digital valuation. We write $\overset{\circ}{D} \in \mathcal{D}_{\underset{\cdot}{\Gamma} \cup \Gamma}$ to denote the digital valuation $\rho(D)$ where $\rho : \underset{\cdot}{\Gamma} \cup \dot{\Gamma} \leftrightarrows \underset{\cdot}{\Gamma} \cup \Gamma$ is defined by $\rho(\underset{\cdot}{x}) \triangleq \underset{\cdot}{x}, \rho(\dot{x}) \triangleq x$. In a similar way, we define $\underset{\circ}{D} \in \mathcal{D}_{\Gamma \cup \dot{\Gamma}}$.

Recall the definition of **return** of the lazy semantics:

$$\boldsymbol{w}\,\langle \nu, \mu \rangle\,\langle \mu', \lambda \rangle \to \boldsymbol{w}\,\langle \nu', \lambda \rangle \quad \text{where } \langle \nu, \mu \rangle \preccurlyeq \langle \nu', \mu' \rangle.$$

We encode this transition as:

1. Let $D_1, D_2 \in \mathcal{D}_{\underset{\cdot}{\Gamma} \cup \dot{\Gamma}}$ such that $\langle \nu, \mu \rangle \models D_1$ and $\langle \mu', \lambda \rangle \models D_2$.
2. Evolving $D_1$ to $\overset{\circ}{D}'_1$ for $\overset{\circ}{D}'_1 \parallel \underset{\circ}{D}_2$ corresponds to $\langle \nu, \mu \rangle \preccurlyeq \langle \nu', \mu' \rangle$.
3. Join and obtain $D \in \overset{\circ}{D}'_1 \oplus \underset{\circ}{D}_2$, then $D \in \mathcal{D}_{\underset{\cdot}{\Gamma} \cup \Gamma \cup \dot{\Gamma}}$ and $\underset{\cdot}{\nu'} \oplus \mu' \oplus \dot{\lambda} \models D$.
4. Removing a middle $D' = D|_{\underset{\cdot}{\Gamma} \cup \dot{\Gamma}}$, and $\langle \nu', \lambda \rangle \models D'$.

In summary, we define the digitized semantics $\mathcal{S}_{\mathbf{digi}}$ as follows.

**Definition 12 (Digitized Semantics $\mathcal{S}_{\mathbf{digi}}$).** A configuration is a pair $\langle q, \boldsymbol{W} \rangle$ of a location $q$ and a stack $\boldsymbol{W} \in \mathcal{D}^*_{\underset{\cdot}{\Gamma} \cup \dot{\Gamma}}$. Hence the set of configurations $\mathit{Conf}(\mathcal{S}_{\mathbf{digi}})$ is $Q \times \mathcal{D}^*$. In the following, we write $D(x) = v$ if $(x, v) \in c_0$ for $D = c_0\, c_1 \ldots c_n$.

For $\tau \in \mathit{Act}$, we define the action $\boldsymbol{W} \xrightarrow{\tau} \boldsymbol{W'}$ by case analysis on $\tau$ as follows:

**Case time:** $\boldsymbol{W} D \to \boldsymbol{W} D'$ where $D \vdash^* D'$.
**Case push:** $\boldsymbol{W} D_1 \to \boldsymbol{W} D'_1 D_2$ where $D'_1 = D_1[\dot{\complement} := 0]$, $\overset{\circ}{D}'_1 \parallel \underset{\circ}{D}_2$, $D_2(\dot{x}) = 0$ $(\forall x \in \Gamma)$.
**Case return:** $\boldsymbol{W} D_1 D_2 \to \boldsymbol{W}(D|_{\underset{\cdot}{\Gamma} \cup \dot{\Gamma}})$ where $D_1 \vdash^* D'_1$, $D \in \overset{\circ}{D}'_1 \oplus \underset{\circ}{D}_2$.
**Case dig**$(x, y)$**:** $\boldsymbol{W} D_1 D_2 \to \boldsymbol{W} D_1(D_2[\dot{x} := y])$.
**Case upd**$(X, \varphi)$**:** $\boldsymbol{W} D \to \boldsymbol{W} D'$ where $D' \in D[\dot{X} \leftarrow \dot{\varphi}]$. ∎

It should be noted that the marked digital valuations over $\underset{\cdot}{\Gamma} \cup \dot{\Gamma}$ are basically the same as the regions considered by Abdulla et al.

We define the well-formedness of a stack, $\mathsf{WF}(\boldsymbol{W})$, inductively:

$\mathsf{WF}(D)$, $\quad \mathsf{WF}(\boldsymbol{W} D_1 D_2)$ if $D_1(\dot{\complement}) = 0$, $D_1 \vdash^* D'_1$, $\overset{\circ}{D}'_1 \parallel \underset{\circ}{D}_2$, and $\mathsf{WF}(\boldsymbol{W} D_1)$.

Every transition preserves well-formedness: if $\mathsf{WF}(\boldsymbol{W})$ and $\boldsymbol{W} \to \boldsymbol{W'}$, then $\mathsf{WF}(\boldsymbol{W'})$.

### 5.1 The Correspondence between $\mathcal{S}_{\mathbf{lazy}}$ and $\mathcal{S}_{\mathbf{digi}}$

We extend the abstraction $\alpha(\boldsymbol{w}) \in D^n_{\underset{\cdot}{\Gamma} \cup \dot{\Gamma}}$ for $\boldsymbol{w} \in (\Lambda \times \Lambda)^n$ inductively as follows:

$$\alpha(\epsilon) \triangleq \epsilon, \qquad \alpha(\boldsymbol{w}\langle \nu, \mu \rangle) \triangleq \alpha(\boldsymbol{w})\,\alpha(\langle \nu, \mu \rangle).$$

**Proposition 13.** If $\boldsymbol{w}$ is well-formed, then $\alpha(\boldsymbol{w})$ is also well-formed.
If $\boldsymbol{W}$ is well-formed, then there exists a well-formed stack $\boldsymbol{w}$ with $\alpha(\boldsymbol{w}) = \boldsymbol{W}$.

**Theorem 14.** The extended abstraction derives the simulation properties.

1. Let $\boldsymbol{w}$ be a well-formed stack of $\mathcal{S}_{\mathbf{lazy}}$. If $\boldsymbol{w} \xrightarrow{\tau} \boldsymbol{w'}$, then $\alpha(\boldsymbol{w}) \xrightarrow{\tau} \alpha(\boldsymbol{w'})$.
2. Let $\boldsymbol{W}$ be a well-formed stack of $\mathcal{S}_{\mathbf{digi}}$. If $\boldsymbol{W} \xrightarrow{\tau} \boldsymbol{W'}$, $\alpha(\boldsymbol{w'}) = \boldsymbol{W'}$, and $\mathsf{WF}(\boldsymbol{w'})$, then there exists $\boldsymbol{w}$ such that $\boldsymbol{w} \xrightarrow{\tau} \boldsymbol{w'}$, $\mathsf{WF}(\boldsymbol{w})$, and $\alpha(\boldsymbol{w}) = \boldsymbol{W}$.

Consider the following forward simulation instead of the later of Theorem 14:

$2'$. Let $\boldsymbol{w}$ and $\boldsymbol{W}$ be well-formed stacks. If $\alpha(\boldsymbol{w}) = \boldsymbol{W}$ and $\boldsymbol{W} \xrightarrow{\tau} \boldsymbol{W'}$, then there exists $\boldsymbol{w'}$ such that $\boldsymbol{w} \xrightarrow{\tau} \boldsymbol{w'}$ and $\alpha(\boldsymbol{w'}) = \boldsymbol{W'}$.

Then there is a counter-example, which is caused by the ambiguity appeared in the **return** transition.

$$\frac{\{\dot{\complement} \mapsto 0, \dot{x} \mapsto 0.2, \underset{\cdot}{x} \mapsto 1.6\}}{\{\dot{\complement} \mapsto 0, \underset{\cdot}{x} \mapsto 0.3, \dot{x} \mapsto 1.6\}_{\boldsymbol{w}}} \models \frac{\{(\dot{\complement}, 0)\}\{(\dot{x}, 0)\}\{(\underset{\cdot}{x}, 1)\}}{\{(\dot{\complement}, 0)\}\{(\underset{\cdot}{x}, 0)\}\{(\dot{x}, 1)\}_{\boldsymbol{W}}} \xrightarrow{\textsf{return}} \emptyset\,\{(\underset{\cdot}{x}, 0)\}\,\{(\dot{x}, 0)\} \quad .$$

Then $\boldsymbol{w} \xrightarrow{\textsf{return}} \left\{\dot{x} \mapsto 0.2, \underset{\cdot}{x} \mapsto 0.3\right\}$ and this is the only valuation obtained from $\boldsymbol{w}$. But, it does not realize $\boldsymbol{W'}$. On the other hand, Abdulla et al. introduced the flattening operator of the stack of regions and gave the elaborated proof of a statement similar to the statement $2'$.

*The role of the reference clock.* The reference clock plays an important role in the proof of Theorem 14. See the following counter-example without the reference clocks:

$$\frac{\emptyset\,\{(\underset{\cdot}{x}, 1)\}\,\{(\dot{x}, 0)\}}{\emptyset\,\{(\underset{\cdot}{x}, 0)\}\,\{(\dot{x}, 1)\}_{\boldsymbol{W}}} \xrightarrow{\textsf{time}} \frac{\emptyset\,\{(\dot{x}, 1)\}\,\{(\underset{\cdot}{x}, 1)\}}{\emptyset\,\{(\underset{\cdot}{x}, 0)\}\,\{(\dot{x}, 1)\}_{\boldsymbol{W'}}} \not\models \frac{\{\dot{x} \mapsto 1.4, \underset{\cdot}{x} \mapsto 1.9\}}{\{\underset{\cdot}{x} \mapsto 0.3, \dot{x} \mapsto 1.6\}_{\boldsymbol{w'}}}$$

There is no *well-formed* stack $\boldsymbol{w}$ such that $\boldsymbol{w} \xrightarrow{\textsf{time}} \boldsymbol{w'}$ and $\alpha(\boldsymbol{w}) = \boldsymbol{W}$ because we lost the ordering information between $(\underset{\cdot}{x}, 0)$ and $(\dot{x}, 0)$ of $\boldsymbol{W}$ in $\boldsymbol{W'}$. Put differently, the separating role of $(\dot{x}, 1)$ and $(\underset{\cdot}{x}, 1)$ is missed in $\boldsymbol{W'}$. However, if we insert the reference clocks into the head position, then the **time**-transition preserves the ordering information:

$$\frac{\{(\underset{\cdot}{\complement}, 0)\}\,\{(\underset{\cdot}{x}, 1)\}\,\{(\dot{x}, 0)\}}{\{(\dot{\complement}, 0)\}\,\{(\underset{\cdot}{x}, 0)\}\,\{(\dot{x}, 1)\}_{\boldsymbol{W}}} \xrightarrow{\textsf{time}} \frac{\emptyset\,\{(\dot{x}, 1)\}\,\{(\underset{\cdot}{\complement}, 0)\}\,\{(\underset{\cdot}{x}, 1)\}}{\{(\dot{\complement}, 0)\}\,\{(\underset{\cdot}{x}, 0)\}\,\{(\dot{x}, 1)\}_{\boldsymbol{W'}}}$$

Then it can be verified that if $\boldsymbol{w'} = \langle \nu, \mu \rangle \langle \mu', \lambda \rangle \models \boldsymbol{W'}$ and $\{\lambda(x)\} \geq \{\mu(x)\}$ then $\boldsymbol{w'}$ is not well-formed. The following lemma is crucial in the proof of Theorem 14 and the reference clock is important to show this.

**Lemma 15.** Let $\theta, \vartheta, \eta \in \Lambda_{\Gamma \cup \dot{\Gamma}}$ be valuations such that $\alpha(\theta) \vdash^* \alpha(\vartheta) \vdash^* \alpha(\eta)$, $\theta \preccurlyeq \eta$, and $\vartheta \preccurlyeq \eta$. If $\theta(\dot{\complement}) = 0$, then $\theta \preccurlyeq \vartheta$.

**Solving Reachability Problem.** We consider the reachability problem from $q_{\text{init}}$ to $q_{\text{final}}$ of SRTA. The problem can be solved by considering the corresponding problem in the digitized semantics as follows.

If there exists a computation from $\langle q_{\text{init}}, 0_\Lambda \rangle$ to $\langle q_{\text{final}}, \omega \rangle$, then there exists one from $\langle q_{\text{init}}, \langle 0_\Lambda, 0_\Lambda \rangle \rangle$ to $\langle q_{\text{final}}, \boldsymbol{w} \rangle$ with $\omega = \Phi(\boldsymbol{w})$ by Theorem 6. Then there exists one from $\langle q_{\text{init}}, \alpha(\langle 0_\Lambda, 0_\Lambda \rangle) \rangle$ to $\langle q_{\text{final}}, \boldsymbol{W} \rangle$ with $\alpha(\boldsymbol{w}) = \boldsymbol{W}$ by Theorem 14.

Conversely, if there exists a computation from $\langle q_{\text{init}}, \alpha(\langle 0_\Lambda, 0_\Lambda \rangle) \rangle$ to $\langle q_{\text{final}}, \boldsymbol{W'} \rangle$ for some stack $\boldsymbol{W'}$, then there exists one from $\langle q_{\text{init}}, \langle 0_\Lambda, 0_\Lambda \rangle \rangle$ to $\langle q_{\text{final}}, \boldsymbol{w'} \rangle$ with $\alpha(\boldsymbol{w'}) = \boldsymbol{W'}$. Then there exists one from $\langle q_{\text{init}}, 0_\Lambda \rangle$ to $\langle q_{\text{final}}, \omega' \rangle$ with $\omega' = \Phi(\boldsymbol{w'})$. We used Theorem 14 and 6 in this order. The choice of $0_\Lambda$ is important because there is no $\langle \nu, \mu \rangle$ such that $\langle \nu, \mu \rangle \neq \langle 0_\Lambda, 0_\Lambda \rangle$ and $\langle \nu, \mu \rangle \models \alpha(\langle 0_\Lambda, 0_\Lambda \rangle)$.

**Solving Reachability Problem of SRTA.** Let $q_{\text{init}}, q_{\text{final}}$ be control locations of SRTA. The reachability problem from $q_{\text{init}}$ to $q_{\text{final}}$ can be solved as the above discussion. Since the digitized semantics $\mathcal{S}_{\mathbf{digi}}$ can be implemented by a pushdown system and the reachability problem of pushdown systems is decidable [6, 9], we can solve the reachability problem.

## 6 Conclusion and Future Work

We have introduced the three semantics (collapsed, lazy, and digitized semantics) for our new model, SRTA, and established the correspondences among these semantics. This step-wise construction reveals the essence of regions considered by Abdulla et al. As a result, our proofs are directly derived from the basic correspondence between collapsed valuations and digital valuations. Since the digitized semantics is finite and can be considered as pushdown systems, the reachability problem of SRTA is decidable. As a corollary, the decidability of the reachability problem of TPDA is proved again in a simpler manner.

As future work, considering two extensions is natural, (i) Adopting the local time elapse transition rather than our **time** transition. Then, is the reachability problem still decidable? Are there any relations about expressiveness between this and ours? (ii) Can we add the diagonal constraints into synchronized recursive timed automata within the reachability problem is still decidable? In the theory of timed automata, it is well known that a timed automaton with diagonal constraints can be encoded into a diagonal-free timed automaton [3, 5]. We believe our new formulation is good clue to investigate these problems.

## References


1. Abdulla, P., Atig, M., Stenman, J.: Dense-timed pushdown automata. In: LICS '12. pp. 35–44. IEEE (2012)
2. Abdulla, P., Čerāns, K., Jonsson, B., Tsay, Y.: Algorithmic analysis of programs with well quasi-ordered domains. Inf. & Comp. 160(1–2), 109–127 (2000)
3. Alur, R., Dill, D.: A theory of timed automata. TCS. 126(2), 183–235 (1994)
4. Benerecetti, M., Minopoli, S., Peron, A.: Analysis of Timed Recursive State Machines. In: TIME. pp. 61–68. IEEE (2010)
5. Bérard, B., Diekert, V., Gastin, P., Petit, A.: Characterization of the expressive power of silent transitions in timed automata. Fundamenta Informaticae 36(2), 145–182 (1998)
6. Bouajjani, A., Esparza, J., Maler, O.: Reachability analysis of pushdown automata: Application to model-checking. In: CONCUR '97, LNCS, vol. 1243, pp. 135–150. Springer (1997)
7. Cai, X., Ogawa, M.: Well-structured pushdown system: Case of dense timed pushdown automata. In: FLOPS '14, LNCS, vol. 8475, pp. 336–352. Springer (2014)
8. Finkel, A., Schnoebelen, P.: Well-structured transition systems everywhere! TCS. 256(1–2), 63–92 (2001)
9. Finkel, A., Willems, B., Wolper, P.: A direct symbolic approach to model checking pushdown systems. In: INFINITY '97, ENTCS, vol. 9, pp. 27–37. Elsevier (1997)
10. Trivedi, A., Wojtczak, D.: Recursive timed automata. In: ATVA '10, LNCS, vol. 6252, pp. 306–324. Springer (2010)