Decidability of Safety Properties of Timed Multiset Rewriting

Mitsuharu Yamamoto[1], Jean-Marie Cottin[2], and Masami Hagiya[2]

[1] Faculty of Science, Chiba University
Yayoicho 1–33, Inage-ku, Chiba, Japan
`mituharu@math.s.chiba-u.ac.jp`

[2] Graduate School of Information Science and Technology, University of Tokyo
Hongo 7–3–1, Bunkyo-ku, Tokyo, Japan
`{jcottin,hagiya}@is.s.u-tokyo.ac.jp`

Abstract. We propose timed multiset rewriting as a framework that subsumes timed Petri nets and timed automata. In timed multiset rewriting, which extends multiset rewriting, each element of a multiset has a clock, and a multiset is transformed not only by usual rewriting but also by time elapse. Moreover, we can specify conditions on clocks for rewriting.

In this paper, we analyze reachability, boundedness, and coverability of timed multiset rewriting. Decidability of each property on the system depends on the presence of invariant rules and diagonal constraints. First, we show that all three properties are undecidable for systems with invariant rules. Then we show that reachability is undecidable, and both boundedness and coverability are decidable for the system without invariant rules. Finally, we show that all the three properties are undecidable if we include diagonal constraints even when excluding invariant rules.

Keywords: real-time systems, timed Petri nets, timed automata, decidability.

1 Introduction

Rewriting on multisets is a framework that can naturally model dynamic creation and destruction of objects or processes [1,2]. In order to model lifetime or timeouts of objects, a number of frameworks that can handle real-time behavior have been proposed [3,4,5,6,7]. As well as rewriting on multisets, Petri nets can model changes in the number of processes, and there are many ways to extend Petri nets so that they can provide real-time features [8].

Automata, which are used as a basis of automatic verification also have some real-time extensions [9,10]. In particular, timed automata have been studied by many researchers and automatic analysis methods using regions or zones are well-established.

In this paper, we propose timed multiset rewriting, which is a real-time extension of multiset rewriting, in a way that can naturally handle clocks as in timed automata. In timed multiset rewriting, each element has its own clock, which gives us expressive power. Moreover, it can be seen as an extension of timed Petri nets.

Among properties on timed multiset rewriting, we focus on reachability, boundedness, and coverability, all of which are naturally imported from properties on Petri nets, and discuss their decidability and analysis methods. In timed multiset rewriting, we can obtain some variations by restricting or allowing various kinds of rules or constraints. In this paper, we analyze decidability of reachability, boundedness, and coverability of timed multiset rewriting, depending on whether invariant rules and diagonal constraints are allowed or not. Decidable properties can be analyzed using regions and zones, which have been used in the analysis of timed automata, and Karp-Miller trees, which have been used in the analysis of Petri nets.

This paper is organized as follows. Section 2 provides the definition of timed multiset rewriting and shows that it is an extension of both timed automata and timed Petri nets. Sections 3 and 4 discuss reachability, boundedness, and coverability of timed multiset rewriting with and without invariant rules, respectively. The effect of diagonal constraints is discussed in Section 5. Finally, we summarize the paper in Section 6.

2 Preliminaries

In this section, we give the definition of timed multiset rewriting, and show that it is an extension of two widely studied classes of real-time systems: timed Petri nets and timed automata.

2.1 Timed Multiset Rewriting

Definition 1. *We write the set of all non-negative real numbers as $\mathbb{R}_{\geq 0}$. Let S be a finite set of elements, and T a set of clock variables. A* pattern *is defined as a finite multiset*

$$M = \{a_1{:}t_1,\, a_2{:}t_2, \ldots,\, a_n{:}t_n\}$$

(where $a_i \in S$, $t_i \in \mathbb{R}_{\geq 0} \cup T$). Since it is a multiset, it may be the case that $a_i = a_j$ or $t_i = t_j$ for $i \neq j$. For a pattern M above, the set $\mathrm{Var}(M)$ *of clock variables, the multiplicity $|M|$, and the multiplicity $|M|_a$ of an element a are defined as follows:*

$$\mathrm{Var}(M) \overset{\mathrm{def}}{=} \{t_i \mid t_i \in T\} \qquad |M| \overset{\mathrm{def}}{=} n$$
$$|M|_a \overset{\mathrm{def}}{=} \textit{the number of elements } a_i \textit{ such that } a_i = a$$

When $\mathrm{Var}(M) = \emptyset$*, we call M a* timed multiset*.*

For an element $a{:}t$ of a pattern, we simply call a an element*, call $t \in \mathbb{R}_{\geq 0} \cup T$ a* clock*, and $a{:}t$ itself a* timed element*.*

We can consider a substitution σ : $\mathrm{Var}(M) \to \mathbb{R}_{\geq 0}$ *from clock variables in a pattern M to non-negative reals. The result of applying a substitution σ to M becomes a timed multiset, and we denote it by $M\sigma$.*

Let $\mathcal{B}(T)$ be the set of constraints whose element is a conjunction made from relations $t \bowtie c\,(t \in T,\ c \in \mathbb{N},\ \bowtie\, \in \{=, \leq, \geq, <, >\})$ between a clock variable and a natural number. When σ satisfies a constraint $\mathcal{C} \in \mathcal{B}(T)$, we write $\sigma \models \mathcal{C}$.

As its name implies, timed multiset rewriting, which is the main target of this paper, is defined as rewriting on timed multisets as defined above.

Definition 2. *A* timed multiset rewriting system *(TMSRS) is given by a tuple $\mathcal{R} = (S, I, J)$, where S is a set of elements, and I and J are sets of invariant rules and jump rules, respectively.*

Each invariant rule *has the following form:*

$$l \mid \mathcal{C}$$

where l is a non-empty pattern whose clocks are clock variables (no non-negative reals), and $\mathcal{C} \in \mathcal{B}(\mathrm{Var}(l))$ is a constraint on clock variables occurring in l. We say that a timed multiset M satisfies invariants I if

$$\forall \sigma.\, l\sigma \subseteq_m M \Longrightarrow \sigma \models \mathcal{C} \ \textit{ for all } l \mid \mathcal{C} \textit{ in } I$$

where $\subseteq_m$ denotes multiset inclusion.

Each jump rule *has the following form:*

$$R:\ l \to r\ \textbf{\textit{if}}\ \mathcal{C}$$

where R is a label stuck to the rule and occurs only once in J, l is a non-empty pattern, r is a possibly empty pattern, and $\mathcal{C} \in \mathcal{B}(\mathrm{Var}(l) \cup \mathrm{Var}(r))$ is a constraint on clock variables occurring in either l or r. Each clock in l or r must be either a natural number or a clock variable.

When writing a pattern in I or J, we omit the surrounding "{" and "}".

Rewriting in a TMSRS $\mathcal{R}$ is classified into two kinds: jump *and* flow*. Jump corresponds to jump rules. For a jump rule $R:\ l \to r$ **if** $\mathcal{C}$ and a timed multiset M satisfying invariants I, if there exist timed multisets M' and N, and a substitution σ satisfying*

$$M = l\sigma + N, \quad M' = r\sigma + N, \quad \sigma \models \mathcal{C}, \quad M' \textit{ satisfies } I$$

(where $+$ denotes multiset union), then we say that M is rewritable to M' and write $M \xrightarrow{(R,\sigma)} M'$ or simply $M \xrightarrow{R} M'$.

The other kind of rewriting, flow, corresponds to passage of time. For a timed multiset $M = \{a_1{:}t_1, \ldots, a_n{:}t_n\}$ satisfying invariants I and a positive real $d \in \mathbb{R}_{>0}$, we define $M + d$ as $M + d \stackrel{\mathrm{def}}{=} \{a_1{:}t_1{+}d, \ldots, a_n{:}t_n{+}d\}$. If $M + d$ satisfies invariants I, then we say M is rewritable to $M' = M + d$ and write $M \xrightarrow{d} M'$.

When M is rewritable to M' using either jump or flow, we write $M \to M'$. The reflexive transitive closure and transitive closure of $\to$ are written $\to^*$ and $\to^+$, respectively.

2.2 Relation with Timed Petri Nets

A timed Petri net [8] is a Petri net augmented by the notion of age. Each token has its own age, and each arc has a time interval.

Definition 3. *A* timed Petri net *is given by a tuple $N = (P, T, In, Out)$ where P is a finite set of places, T is a finite set of transitions ($P \cap T = \emptyset$), $In, Out : T \times P \to \mathcal{M}(\mathbb{N} \times (\mathbb{N} \cup \{\infty\}))$ are functions each of which takes a pair of a transition and a place and returns a finite multiset of intervals.*

Since each token has its own age, a marking M in a timed Petri net is represented by a function from a place P to a multiset $\mathcal{M}(\mathbb{R}_{\geq 0})$ of non-negative reals. In the initial marking, each token is assumed to have its age 0.

Transitions in timed Petri nets are classified into two kinds: those by firing as in the usual Petri nets, and those by passage of time. A marking M is fireable at a transition only if the age of each token (corresponding to the transition) is in the interval given by In. After firing, new tokens are created so that their ages satisfy the interval specified by Out. The transition by passage of time increases all of the ages of tokens by a positive real d.

A timed Petri net can be simulated by a TMSRS as follows.

Definition 4. *For a timed Petri net $N = (P, T, F, W, times)$, we define a TMSRS $\mathcal{R} = (S, I, J)$ as follows:*

- *$S = P$ and $I = \emptyset$*
- *For each $\tau \in T$, J has the following rule:*

$$\begin{array}{c} R_\tau : p_1{:}t_1^1, \ldots, p_1{:}t_1^{w_1}, \ldots, p_n{:}t_n^1, \ldots, p_n{:}t_n^{w_n} \to \\ q_1{:}s_1^1, \ldots, q_1{:}s_1^{v_1}, \ldots, q_m{:}s_m^1, \ldots, q_m{:}s_m^{v_m} \\ \textbf{\textit{if}}\ c_1^1 \wedge \cdots \wedge c_1^{w_1} \wedge \cdots \wedge c_n^1 \wedge \cdots \wedge c_n^{w_n} \wedge \\ d_1^1 \wedge \cdots \wedge d_1^{v_1} \wedge \cdots \wedge d_n^1 \wedge \cdots \wedge d_m^{v_m} \end{array}$$

  where

  $\{p_1, \ldots, p_n\} = \{p \mid In(\tau, p) \neq \emptyset\}$, $w_i = |In(\tau, p_i)|$,
  $\{q_1, \ldots, q_m\} = \{q \mid Out(\tau, q) \neq \emptyset\}$, $v_i = |Out(\tau, q_i)|$,

  $$\textit{If } In(\tau, p_i) = \{(l_1, u_1), \ldots, (l_{w_i}, u_{w_i})\}, \begin{cases} c_i^j = l_j \leq t_i^j & \textit{if } u_j = \infty \\ c_i^j = l_j \leq t_i^j \wedge t_i^j \leq u_j & \textit{otherwise} \end{cases},$$

  $$\textit{If } Out(\tau, q_i) = \{(l_1, u_1), \ldots, (l_{v_i}, u_{v_i})\}, \begin{cases} d_i^j = l_j \leq s_i^j & \textit{if } u_j = \infty \\ d_i^j = l_j \leq s_i^j \wedge s_i^j \leq u_j & \textit{otherwise} \end{cases}$$

  We assume t_i^j and s_i^j are clock variables in $\mathcal{R}$.
- *The initial timed multiset M_0 is a multiset of tokens in the initial marking with all their clocks 0.*

The close relation between TMSRSs and timed Petri nets naturally induces properties on TMSRSs that are originally defined on Petri nets, and they can be used for safety analysis, i.e., proving bad things never happen starting from a certain state.

Definition 5. *Assume that a TMSRS $\mathcal{R}$ and an initial timed multiset M_0 are given. A timed multiset M is* reachable *from M_0 if $M_0 \to^* M$. For a timed multiset M', M' is* coverable *from M_0 if there exists M such that $M_0 \to^* M$ and $M' \subseteq_m M$. $\mathcal{R}$ is* bounded *from M_0 if a timed multiset reachable from M_0 has a finite upper bound in its multiplicity.*

If we restrict the *Out* function's range to be $\mathcal{M}(\{(0,0)\})$, then we obtain timed-arc Petri nets. For timed-arc Petri nets, the reachability is shown to be undecidable [11], hence it is also undecidable for timed Petri nets. In [12], boundedness and coverability are shown to be decidable for discrete-time timed-arc Petri nets, i.e., the age of each token is always a natural number. The full continuous-time timed Petri nets are analyzed in [13], and their coverability is decidable.

It should be noted that there are differences between timed Petri nets and TMSRSs in their expressiveness. Besides the presence of invariant rules, timed Petri nets can not express a jump rule in which there are multiple occurrences of a single variable. This feature enables us to express preservation of clock values and a limited form of diagonal constraints, which will be discussed in Section 5.

In [7], correspondence between a timed Petri net and a system that operates on multisets is also described in the context of first-order linear logic programs, which can express more broader classes than timed Petri nets. Although [7] uses the result of the coverability of timed Petri nets for the analysis of first-order linear logic programs, it does not extend the decidable class. Thus our coverability result can also extend the decidable class for first-order linear logic programs.

2.3 Relation with Timed Automata

A timed automaton [9] is a timed extension of a usual automaton, by adding several (fixed number of) clocks to the states. It enable us to specify both discrete behavior of control and continuous behavior of time.

Definition 6. *A* timed automaton *is given by a tuple $A = (L, l_0, X, E, Inv)$, where L is a finite set of locations, $l_0 (\in L)$ is the initial location, $E (\subseteq L \times \mathcal{B}(X) \times 2^X \times L)$ is a set of edges, and $Inv : L \to \mathcal{B}(X)$ is an assignment of invariants to locations. In the case of $(l, g, r, l') \in E$, we write $l \xrightarrow{g,r} l'$.*

A state of a timed automaton is represented by a pair (l, u) of a location l and a clock assignment $u : X \to \mathbb{R}_{\geq 0}$. In the initial state, the location l coincides with the initial location l_0, and the clock assignment $u = u_0$ assigns 0 to each clock $x \in X$.

A timed automaton has two kinds of transitions as follows:

- $(l, u) \to (l', u')$ *where* $l \xrightarrow{g,r} l'$, $u \models g$, $u' = [r \mapsto 0]u$, $u' \models Inv(l')$
- $(l, u) \to (l, u + d)$ *if* $u + d \models Inv(l)$

The former is a transition between locations through an edge as in the usual automata, and one can reset the values of the clocks, which are specified in the *edge, to* 0*. The latter is a transition by time elapse without changing its location, and the value of every clock increases uniformly. For both transitions, the destination state* (l, u) *has to satisfy the invariant condition* $Inv(l)$ *that is assigned to the destination location* l*.*

Since the timed automaton defined above contains location invariants, it is actually a *timed safety automaton* [14], rather than the original timed automaton in [9] that has Büchi acceptance conditions instead of location invariants.

As in the case of timed Petri nets, a timed automaton can be simulated by a TMSRS. Both locations and clocks in a timed automaton are represented by clock variables in a TMSRS, permissive time progress at each location by an invariant, and a possible transition by a constraint on clock variables. A similar formulation of timed automata is also described as a special class of *real-time systems* in [15].

Definition 7. *For a given timed automaton* $A = (L, l_0, X, E, Inv)$*, we define a TMSRS* $\mathcal{R} = (S, I, J)$ *as follows.*

- $S = L \cup \{c_x \mid x \in X\}$
- *For each* $l \in L$*,* I *has the following invariant rule:*

  $$l{:}t,\ c_{x_1}{:}x_1, \ldots, c_{x_n}{:}x_n \mid Inv(l)$$

  where x_i *is a clock variable occurring in* $Inv(l)$*. We assume* t *and* x_i *are clock variables in* $\mathcal{R}$*.*
- *For each* $e = (l, g, r, l') \in E$*,* J *has the following jump rule.*

  $$R_e : l{:}t,\ c_{x_1}{:}x_1, \ldots, c_{x_n}{:}x_n \rightarrow l'{:}t,\ c_{x_1}{:}y_1, \ldots c_{x_n}{:}y_n \ \textbf{\textit{if}}\ g$$

  where x_i *is a clock variable occurring in either* g *or* r*, and* y_i *is* 0 *if* $x_i \in r$*, otherwise* x_i*. We assume* t *and* x_i *are clock variables in* $\mathcal{R}$*.*
- *The initial timed multiset* M_0 *is given by* $\{l_0{:}0\} \cup \{c_x{:}0 \mid x \in X\}$*.*

In timed automata, both reachability and coverability are decidable since we can enumerate all reachable states using *region*s. On the other hand, boundedness is meaningless since the multiplicity does not change through the transitions.

3 With Invariant Rules

This section is devoted to the undecidability results for reachability, boundedness, and coverability of the TMSRS given in the previous section. We consider the full version of a TMSRS including invariant rules in this section, and the next section is for a TMSRS without invariant rules.

3.1 Simulating 2-Counter Machines

The undecidability of reachability, boundedness, and coverability of a TMSRS are shown by using the fact that the halting problem of 2-counter machines is undecidable.

Definition 8. *A 2-counter machine is given by a sequence $I_1, \dots, I_n$ of instructions, which are operations on two counters r_1 and r_2.*

The execution of a 2-counter machine starts from I_1. There is an instruction I_e that designates the end of the execution, and when the execution reaches the instruction I_e, the execution terminates. Each instruction has either of the following two types.

– *Increment: $I(i, j, k)$*

$$I_j : r_i := r_i + 1;\ \textbf{\textit{goto}}\ I_k$$

– *Test&Decrement: $D(i, j, k, l)$*

$$I_j : \textbf{\textit{if}}\ r_i > 0\ \textbf{\textit{then}}\ r_i := r_i - 1;\ \textbf{\textit{goto}}\ I_k\ \textbf{\textit{else goto}}\ I_l$$

Theorem 1. *The halting problem of 2-counter machines is undecidable.*

Next, we simulate 2-counter machines by a TMSRS. The method shown here is based on the simulation of 2-counter machines by timed-arc Petri nets given in [11], where undecidability of reachability of timed-arc Petri nets is shown using the simulation. Compared with [11], our simulation is strong enough to show undecidability of boundedness and coverability, as well as reachability, using invariant rules.

Definition 9. *We associate an element p_j for each instruction I_j in a 2-counter machine. The multiplicity of p_j is assumed to be at most 1, and it corresponds to the execution of the instruction I_j. Moreover, we associate an element r_i for each counter r_i in a 2-counter machine. The multiplicity of the timed element $r_i{:}1$ stands for the value of r_i in a 2-counter machine.*

For each type of instruction, we associate the following jump rules.

– *Increment: $I(i, j, k)$*

$$I_j : p_j{:}1 \to p_k{:}1,\ r_i{:}1$$

– *Test&Decrement: $D(i, j, k, l)$*

$$\begin{array}{ll} I_j^1 : p_j{:}1,\ r_i{:}1 \to p_k{:}1 & \quad I_j^3 : r_{3-i}{:}2,\ q_j{:}0 \to r_{3-i}{:}0,\ q_j{:}0 \\ I_j^2 : p_j{:}2 \to q_j{:}0 & \quad I_j^4 : q_j{:}0 \to p_l{:}0 \end{array}$$

where r_{3-i} is the counter that is not examined whether its value is 0 or not.

Moreover we add the following invariant rules:

$$r_i{:}t \mid t \le 2 \quad (i = 1, 2)$$

A state in a 2-counter machine is represented by the following timed multiset:

$$\{p_j{:}1, \overbrace{r_1{:}1, \dots, r_1{:}1}^{n_1}, \overbrace{r_2{:}1, \dots, r_2{:}1}^{n_2}\}$$

For brevity, the above timed multiset is denoted by (j, n_1, n_2).

The initial multiset is given by $(1, n_1, n_2)$, if the initial value of the counter r_i is n_i in a 2-counter machine.

Lemma 1. *Let CM be a 2-counter machine, and $\mathcal{R}$ a TMSRS given by Definition 9. Then for a timed multiset $M = (j, n_1, n_2)$, there exists a sequence of at least 1 rewrite starting from M and reaching the timed multiset of the form (j', n'_1, n'_2). Moreover, for such a multiset (j', n'_1, n'_2), if the instruction at I_j is Increment ($I(i, j, k)$), we have*

$$j' = k,\ n'_i = n_i + 1,\ n'_{3-i} = n_{3-i}$$

and if it is Test&Decrement($D(i, j, k, l)$), we have

$$\begin{cases} j' = k,\ n'_i = n_i - 1,\ n'_{3-i} = n_{3-i} & (\text{if } n_i > 0) \\ j' = l,\ n'_i = 0,\ n'_{3-i} = n_{3-i} & (\text{if } n_i = 0) \end{cases}$$

Proof. 1. Case where the instruction I_j is Increment $I(i, j, k)$: We only consider the case $i = 1$, but the case $i = 2$ is similar. The only applicable jump rule is I_j and we have $(j, n_1, n_2) \xrightarrow{I_j} (k, n_1 + 1, n_2)$.
If we apply flow, the clock assigned to p_j exceeds 1, and no more jumps are applicable. So no further rewriting reaches (j', n'_1, n'_2).
2. Case where the instruction I_j is Test&Decrement $D(i, j, k, l)$: As in the case of Increment, we only consider the case $i = 1$. If $n_1 = 0$, then the possible sequence of rewrites is:

$$\{p_j{:}1,\ r_2{:}1, \ldots,\ r_2{:}1\} \xrightarrow{d_1} \cdots \xrightarrow{d_n} \{p_j{:}2,\ r_2{:}2, \ldots,\ r_2{:}2\} \xrightarrow{I_j^2}$$
$$\{q_j{:}0,\ r_2{:}2, \ldots,\ r_2{:}2\} \xrightarrow{I_j^3}{}^{*} \{q_j{:}0,\ r_2{:}t_1, \ldots,\ r_2{:}t_m\} \xrightarrow{I_j^4} \{p_l{:}0,\ r_2{:}t_1, \ldots,\ r_2{:}t_m\}$$

where t_i is either 0 or 2 and $d_1 + \cdots + d_n = 1$. If t_i is 0 for $i = 1, \ldots, m$, then we can rewrite to $(l, 0, n_2)$ by flow. Otherwise, no jump rule is applicable, and neither is flow because of invariant rules.
If $n_1 > 0$, we have two possibilities: rewrite to $(k, n_1 - 1, n_2)$ with the jump rule I_j^1, or to

$$\{p_j{:}2,\ r_1{:}2, \ldots,\ r_1{:}2,\ r_2{:}2, \ldots,\ r_2{:}2\}$$

with I_j^2 after 1 time unit. In the latter, we can further apply I_j^3 several times and then I_j^4, but at least one $r_1{:}2$ remains. Hence no further rewriting is applicable.

□

Theorem 2. *Let CM be a 2-counter machine, and $\mathcal{R}$ be the corresponding TMSRS. Then the following two conditions are equivalent.*

1. *If we execute CM from the state for which the initial value of r_i is n_i, then it terminates at the state for which the value of r_i is m_i.*
2. *In $\mathcal{R}$, $(1, n_1, n_2) \rightarrow^* (e, m_1, m_2)$.*

3.2 Reachability, Boundedness, and Coverability

The undecidability of reachability of a TMSRS is a direct consequence of that of timed-arc Petri nets [11]. However, we can also easily show it using the simulation given in Definition 9.

In general, the values of counters r_1 and r_2 are arbitrary when a 2-counter machine reaches I_e. However, without loss of generality, we can consider an extended machine that the final values of counters become 0 whenever it terminates, by adding appropriate instructions after I_e. We use the extended machine in the arguments on the reachability of a TMSRS.

Theorem 3. *The reachability of a TMSRS is undecidable.*

Proof. By Theorem 2, for an extended 2-counter machine CM, the initial values n_1 and n_2 and the corresponding TMSRS $\mathcal{R}$, the termination of CM and the existence of a sequence of rewrites $(1, n_1, n_2) \to^* (e, 0, 0)$ in $\mathcal{R}$ is equivalent. Since the halting problem of 2-counter machines is undecidable, the reachability of a TMSRS is also undecidable.

As in reachability, the undecidability of boundedness of a TMSRS is shown by that of the halting problem of 2-counter machines. First, we eliminate the jump rules corresponding to the instruction I_e, so that we may not apply jump rules after reaching I_e.

Since we represent the value of a counter by a multiplicity of the corresponding timed element, if a TMSRS $\mathcal{R}$ simulating a 2-counter machine CM is bounded, then the space of possible values of counters is finite. In this case, we can determine whether the machine terminates or not using some exhaustive search. Therefore, if we can show that

$$\mathcal{R} \text{ is unbounded} \Longrightarrow CM \text{ does not terminate},$$

then the boundedness of the TMSRS determines the termination of the 2-counter machine, hence boundedness of a TMSRS is undecidable.

Lemma 2. *For a 2-counter machine CM, let $\mathcal{R}$ be a TMSRS simulating CM (with elimination of the jump rules corresponding to the instruction I_e). Then CM does not terminate whenever $\mathcal{R}$ is unbounded.*

Proof. In order to make $\mathcal{R}$ unbounded, rewriting by jump must occur infinitely often. This corresponds to the infinite execution of CM. □

Theorem 4. *The boundedness of a TMSRS is undecidable.*

Finally, the undecidability of coverability of a TMSRS is shown using the same simulation.

Lemma 3. *Let CM be a 2-counter machine, and $\mathcal{R}$ be the simulating TMSRS. Then CM terminates if and only if $\{p_e{:}1\}$ is coverable in $\mathcal{R}$.*

Proof. A sequence of rewrites reaching a timed multiset that contains $p_e{:}1$ corresponds to the execution of the 2-counter machine that reaches I_e. □

Theorem 5. *The coverability of a TMSRS is undecidable.*

4 Without Invariant Rules

As we mentioned in the previous section, reachability, boundedness, and coverability are all undecidable in general for a TMSRS. In this section, we consider restricted TMSRSs that do not contain invariant rules, and discuss reachability, boundedness, and coverability of those systems. A "TMSRS" in this section is assumed to be one without invariant rules even if we do not state it explicitly.

4.1 Reachability

In Definition 8, the simulation of timed(-arc) Petri nets by a TMSRS does not require invariant rules. Thus the undecidability result for the reachability of timed-arc Petri nets [11] is also applicable to a TMSRS without invariant rules. In fact, the essential part of the argument in [11] can also be imitated in the simulation given in Definition 9.

Theorem 6. *The reachability of a TMSRS without invariant rules is undecidable.*

4.2 Boundedness

Here we show that the boundedness of a TMSRS is decidable. In a TMSRS, there are uncountably many rewrites from a certain timed multiset, and that makes it difficult to analyze a TMSRS directly. Hence we use the notion of regions in order to make rewrites finitely many. The notion of regions was originally introduced on a fixed set of clock variables for analysis of timed automata, and it was extended to that on multisets for reachability analysis of timed processes [16].

Definition 10. *For a given TMSRS $\mathcal{R}$, we denote by C_{M} the maximum natural number occurring in $\mathcal{R}$ as a constant in a constraint.*

The region $M^\star$ corresponding to a timed multiset M is defined by the triple consisting of the following components.

1. *The multiset $\{(a{:}t) \mid a{:}t \in M,\ t \in \mathbb{N},\ t \leq C_{\mathrm{M}}\}$*
2. *The list of multisets $[B_1; B_2; \ldots; B_n]$ defined as follows. Let $0 < f_1 < f_2 < \cdots < f_n$ be all the non-zero fractional parts of clocks in M. Then B_i is defined as the multiset $\{(a{:}\lfloor t \rfloor) \mid a{:}t \in M,\ \mathrm{fr}(t) = f_i,\ t < C_{\mathrm{M}}\}$, where $\lfloor t \rfloor$ and $\mathrm{fr}(t)$ denote the integral and fractional parts of t, respectively.*
3. *The multiset of multisets $\{C_1, \ldots, C_m\}$ defined as follows. Let $I = \{c_i, \ldots, c_m\}$ be a set of natural numbers satisfying $I = \{t \mid a{:}t \in M,\ t > C_{\mathrm{M}}\}$. Then C_i is defined as the multiset $\{a \mid a{:}c_i \in M\}$.*

For a region U, its multiplicity $|U|$ is defined as the number of elements ($(a{:}t)$ or a for all $a \in S$), and the multiplicity $|U|_a$ of an element a is defined as the number of a occurring in U.

Example 1. When $C_{\mathrm{M}} = 3$ and a timed multiset M is

$$\{a{:}2.2,\ a{:}1.2,\ a{:}1.2,\ a{:}1,\ b{:}3.2,\ b{:}3,\ b{:}3,\ c{:}0.9\},$$

then $M^\star$ is

$$(\{(a{:}1),\ (b{:}3),\ (b{:}3)\},\ [\{(a{:}2),\ (a{:}1),\ (a{:}1)\};\ \{(c{:}0)\}],\ \{\{b\}\}).$$

For the above $M^\star$, $|M^\star| = 8$ and $|M^\star|_b = 3$.

Proposition 1. *$|M^\star| = |M|$ and $|M^\star|_a = |M|_a$ for a multiset M and an element $a \in S$.*

Proposition 2. *For a region U, there exists a timed multiset M such that $M^\star = U$.*

Rewriting on regions is defined as well as on timed multisets.

Definition 11. *Let $\mathcal{R}$ be a TMSRS.*

*For a region U and a jump rule $R : l \to r$ **if** $\mathcal{C}$, if there exists a timed multiset N, a region U', and an assignment σ such that*

$$U = (l\sigma + N)^\star, \quad U' = (r\sigma + N)^\star, \quad \sigma \models \mathcal{C},$$

then we say that U is rewritable to U', and write $U \xrightarrow{R} U'$.

A rewrite by flow is defined as follows. For a region $U = (A, [B_1; \ldots; B_n], C)$ $(n \geq 0)$ satisfying either $A \neq \emptyset$ or $n > 0$, a region U' is defined as follows:

1. *Case $A \neq \emptyset$:*

$$U' \stackrel{\mathrm{def}}{=} \begin{cases} (\emptyset, [A'; B_1; \ldots; B_n], C' + C) & \text{if } A' \neq \emptyset \\ (\emptyset, [B_1; \ldots; B_n], C' + C) & \text{if } A' = \emptyset \end{cases}$$

   where $A' \stackrel{\mathrm{def}}{=} \{(a{:}t) \mid (a{:}t) \in A,\ t < C_{\mathrm{M}}\}$, $A'' \stackrel{\mathrm{def}}{=} \{a \mid (a{:}t) \in A,\ t = C_{\mathrm{M}}\}$, and $C' = \emptyset$ if $A'' = \emptyset$ otherwise $C' = \{A''\}$.
2. *Case $A = \emptyset$ and $n > 0$:*

$$U' \stackrel{\mathrm{def}}{=} (B'_n, [B_1; \ldots; B_{n-1}], C)$$

   where $B'_n \stackrel{\mathrm{def}}{=} \{(a{:}t{+}1) \mid (a{:}t) \in B_n\}$.

Then we say that U is rewritable to U' and write as $U \xrightarrow{\mathcal{T}} U'$.

When U is rewritable to U' by either jump or flow, we write $U \to U'$. The reflexive transitive closure (transitive closure, resp.) of $\xrightarrow{\mathcal{T}}$ and $\to$ are denoted by $\xrightarrow{\mathcal{T}}^*$ and $\to^*$ ($\xrightarrow{\mathcal{T}}^+$ and $\to^+$), respectively.

Rewriting relation on timed multisets and that on regions are bisimilar to each other.

Proposition 3. *For a TMSRS $\mathcal{R}$, we have the following properties:*

- $M \overset{(R,\sigma)}{\to} N \Longrightarrow M^\star \overset{R}{\to} N^\star$
- $M \overset{d}{\to} N \Longrightarrow M^\star \overset{\tau}{\to}^* N^\star$
- $M^\star \overset{R}{\to} V \Longrightarrow \exists N.\, M \overset{R}{\to} N \land N^\star = V$
- $M^\star \overset{\tau}{\to} V \Longrightarrow \exists N\, d.\, d \in \mathbb{R}_{>0} \land M \overset{d}{\to} N \land N^\star = V$

Lemma 4. *For a TMSRS $\mathcal{R}$, the boundedness of rewriting on timed multisets from M and that on regions from $M^\star$ are equivalent.*

Proof. If one system is not bounded, then for any natural number n, there exists a sequence of rewrites whose result has multiplicity exceeding n. By Proposition 3, such a sequence is simulated by the other, and the multiplicity of the result also exceeds n. □

By Lemma 4, the boundedness of a TMSRS is determined by that on regions. Therefore, we focus on the decidability of the boundedness of rewriting on regions in the rest of this subsection.

For regions introduced on TMSRSs, we can define an ordering between them that can be naturally derived from multiset inclusion, and this ordering plays a significant role in analysis using regions. This is a distinguishing characteristic of regions on multisets, compared with those defined on timed automata.

Definition 12. *Let $U = (A, B, C)$ and $U' = (A', B', C')$ be regions. We say that U' covers U and write $U \preceq U'$ if the following conditions are satisfied,*

- $A \subseteq_m A'$.
- $B \subseteq_m^* B'$ *where $\subseteq_m^*$ is a 'substring' ordering defined by $\subseteq_m$ as follows:*

$$\begin{aligned}&[M_1; M_2; \ldots; M_n] \subseteq_m^* [M'_1; M'_2; \ldots; M'_{n'}]\\ &\overset{\text{def}}{\Longleftrightarrow} \exists \rho : \{1, \ldots, n\} \to \{1, \ldots, n'\} \textit{ strictly monotone.}\\ &\qquad \forall i \in \{1, \ldots, n\}.\, M_i \subseteq_m M'_{\rho(i)}\end{aligned}$$

- $C \subseteq_{mm} C'$ *where $\subseteq_{mm}$ is defined as follows: $C = \{C_1, \ldots, C_n\} \subseteq_{mm} C' \overset{\text{def}}{\Longleftrightarrow}$ there exists $C'' = \{C'_1, \ldots, C'_n\} \subseteq_m C'$ such that $C_i \subseteq_m C'_i$ for all $i = 1, \ldots, n$.*

Actually, $\preceq$ is a partial ordering. We write $U \prec U'$ if $U \preceq U'$ and $U' \not\preceq U$.

Lemma 5. *If $U \preceq U'$ and $U \to V$, then there exists V' such that $U' \to^+ V'$ and $V \preceq V'$. In particular, $U' \overset{R}{\to} V'$ when $U \overset{R}{\to} V$.*

If a TMSRS contains invariant rules I, Lemma 5 above no longer holds with respect to a rewrite by flow.

Example 2. Assume that a TMSRS has an invariant rule $a{:}t \mid t \leq 2$, and $C_{\mathrm{M}} = 3$. Then a rewrite by flow

$$(\{(a{:}1)\},\ [],\emptyset) \overset{\tau}{\to} (\emptyset,\ [\{(a{:}1)\}],\ \emptyset)$$

is possible. However,

$$(\{(a{:}1),\ (a{:}2)\},\ [],\ \emptyset) \xrightarrow{\tau} (\emptyset,\ [\{(a{:}1),\ (a{:}2)\}],\ \emptyset)$$

is not possible since the right hand side is a region for which the invariant rule does not hold.

Definition 13. *A quasi-ordering $\leq$ on a set X is* well-quasi-ordering *(wqo) if for any sequence $x_1, x_2, \ldots$ of elements in X, there exist $i < j$ such that $x_i \leq x_j$.*

Theorem 7. *The relation $\preceq$ on regions is wqo.*

Proof. There are finitely many $(a{:}t)$'s since t is a natural number bounded by C_{M}. Using the fact that $\leq$ on natural numbers is wqo and a finite product of wqo is also wqo, all $\subseteq_m$ in Definition 12 are wqo. Furthermore, since the 'substring' ordering derived from wqo is also wqo (Higman's lemma), $\subseteq_m^*$ is wqo. Hence $\preceq$ is wqo. □

Proposition 4. *For $\nu : S \to \mathbb{N}$, the regions U satisfying $\forall a.\, |U|_a = \nu(a)$ are finite.*

Proof. Suppose they are infinitely many. Since $|U| = |V|$ and $U \preceq V$ imply $U = V$, it contradicts the fact that $\preceq$ is wqo.

Proposition 5. *For a region U, the set $\{V \mid U \to V\}$ is finite and computable.*

Proof. For a rewrite by flow, Definition 11 directly leads to the computation of the set.

For a rewrite by jump, using Proposition 2, we can take a concrete M such that $M^\star = U$. By Proposition 3, we can collect all regions V such that $U \xrightarrow{R} V$ by collecting $N^\star$ for all multisets N such that $M \xrightarrow{R} N$. For a jump rule $R : l \to r$ **if** $\mathcal{C}$, the set $\{\sigma|_{\mathrm{Var}(l)} \mid \sigma \models \mathcal{C},\ l\sigma \subseteq_m M\}$ of substitutions are finite, where $\sigma|_{\mathrm{Var}(l)}$ is the restriction of the domain of σ to $\mathrm{Var}(l)$. Thus for each substitution $\sigma' : \mathrm{Var}(l) \to \mathbb{R}_{\geq 0}$, the set $G \overset{\text{def}}{=} \{(r\sigma)^\star \mid \sigma \models \mathcal{C},\ \sigma|_{\mathrm{Var}(l)} = \sigma'\}$ of regions are also finite by Proposition 4 since $|(r\sigma_1)^\star|_a = |(r\sigma_2)^\star|_a$ for any element a and substitutions $\sigma_1, \sigma_2 \in G$. Thus the set $\{V \mid U \xrightarrow{R} V\} = \{(r\sigma + L)^\star \mid \sigma \models \mathcal{C},\ M = l\sigma + L\}$ is finite. □

The Karp-Miller tree construction is used to determine whether a Petri net is bounded or not. Here we apply a Karp-Miller tree to a TMSRS, so that it can be used for the boundedness of a TMSRS.

Definition 14. *A Karp-Miller tree for a region U_0 is constructed as follows:*

1. *Attach the label U_0 to the root node s_0. Execute Step 2 with taking the current node as s_0.*
2. *Let s be the current node and U be its label. For each U' such that $U \to U'$ and U' does not coincide with any labels of nodes on the path from the root node to s, create a new node s' as a child of s. The label of s' is defined as follows:*

- *If U' covers the label of some node on the path from the root node to s, then attach the label ∞ to s'.*
- *Otherwise, attach the label U' to s' and execute Step 2 taking s' as the current node.*

Theorem 8. *The Karp-Miller tree for a region U_0 is finite and computable.*

Proof. First, by Proposition 5, $\{V \mid U \to V\}$ is finite and computable for a region U. Suppose the Karp-Miller tree is infinite. Any infinite tree with finite branching contains an infinite path from the root node by König's lemma. By the construction of the Karp-Miller tree, for node s and its ancestor s', if s and s' have labels U and U', respectively, then $U' \not\preceq U$. Thus the existence of such an infinite path contradicts the fact that $\preceq$ is wqo. □

Theorem 9. *For a TMSRS $\mathcal{R}$ without invariant rules, it is bounded from U_0 if and only if the Karp-Miller tree for U_0 does not include the label ∞.*

Proof. If the Karp-Miller tree for U_0 includes a node s with its label ∞, then there exist an ancestor node s' of s. By Lemma 5, we can repeatedly apply a path from s' to s and increase multiplicity as much as we want. Thus $\mathcal{R}$ is unbounded from U_0.

On the other hand, if the Karp-Miller tree for U_0 does not include the label ∞, all the possible rewrites are captured by the Karp-Miller tree. Hence $\mathcal{R}$ is bounded from U_0. □

Therefore, we can determine whether a TMSRS $\mathcal{R}$ is bounded from a timed multiset M by examining whether the Karp-Miller tree for $M^\star$ includes ∞.

Theorem 10. *The boundedness of a TMSRS without invariant rules is decidable.*

4.3 Coverability

The a coverability of TMSRS can be shown by extending the method used in that of timed Petri nets [13], where existential zones are used as an abstract domain of configurations in timed Petri nets. However, the argument in [13] lacks the relationship between configurations and existential zones, and that buries subtle conditions required for the decidability of coverability. So, we first precisely relate multisets, which correspond to configurations in timed Petri nets, and regions as components of existential zones, and then extend the analysis method by existential zones so that it can be applied to analysis of a TMSRS.

Definition 15. *We say that a set K of timed multisets is* total with respect to the closed part of a region $U = (A, B, C)$ *if* $\forall M \in K.\, M^\star = U$, *and* $\{M' \mid M'^\star = (A, B, \emptyset)\} \subseteq \{M_{\leq C_{\mathrm{M}}} \mid M \in K\}$, *where* $M_{\leq C_{\mathrm{M}}} \overset{\mathrm{def}}{=} \{a{:}t \mid a{:}t \in M,\, t \leq C_{\mathrm{M}}\}$.

Lemma 6. *For regions U and V such that $U \to V$, if K is total w.r.t. the closed part of U, then $\{N \mid \exists M \in K.\, M \to N\}$ is also total w.r.t. the closed part of V.*

With the above lemma, two kinds of coverability are related as follows:

Lemma 7. *Assume a TMSRS $\mathcal{R}$ and timed multisets M and N are given. When all the clocks in M are natural numbers, N is coverable from M if and only if $N^\star$ is coverable from $M^\star$, with the proviso that we increase C_{M} enough to make clocks in both M and N smaller than or equal to C_{M} if needed.*

Proof. Taking the greater C_{M} does not affect arguments we have done so far. If we take C_{M} as above, $M^\star$ and $N^\star$ are written $M^\star = (A, B, \emptyset)$ and $N^\star = (A', B', \emptyset)$.

If $N^\star$ is coverable from $M^\star$, then $M^\star \to^* V$ and $N^\star \preceq V$ for some V. Since all the clocks in M are natural numbers and less than or equal to C_{M}, $\{M\}$ is total w.r.t. the closed part of $M^\star$. If we let $V = (A'', B'', C'')$, then we can make N' such that $N'^\star = (A'', B'', \emptyset)$ by adding some timed elements to N. By Lemma 6, there exists N'' such that $N' \subseteq_m N''$, $M \to^* N''$ and $N''^\star = V$. Thus N is coverable from M.

Conversely, if N is coverable from M, then there exists N' such that $M \to^* N'$ and $N \subseteq_m N'$. By Proposition 3, we have $M^\star \to^* N'^\star$ and $N^\star \preceq N'^\star$. That means $N^\star$ is coverable from $M^\star$. □

A similar argument can be applied to the Karp-Miller tree on regions, and we can show that the reachability from a timed multiset whose clocks are natural numbers can be determined by the Karp-Miller tree provided it is bounded (i.e., no ∞ labels).

Although it is possible to show the decidability of the coverability of rewriting on regions using the theory of well-structured transition systems [17], we instead extend the method in [13] using existential zones since it is more concise. The argument on the relationship between two kinds of coverability is also applicable to existential zones since each existential zone can be seen as a union of some regions, and each region can be expressed by some zone. As in [13], we only consider the case that the inequalities occurring in the rewrite rules are non-strict for simplicity, but the general case can be obtained straightforwardly.

Definition 16. *An* existential zone *Z is a triple (m, p, D), where m is a natural number, $p : \{1, \ldots, m\} \to S$ is a mapping from indices to elements, and $D : \{0, \ldots, m\} \times \{0, \ldots, m\} \to \{-C_{\mathrm{M}}, \ldots, C_{\mathrm{M}}\} \cup \{\infty\}$ is a difference bound matrix.*

An existential zone $Z = (m, p, D)$ defines a set $[\![Z]\!]$ of timed multisets as follows:

$$\begin{aligned}
&M = \{a_1{:}t_1, \ldots, a_n{:}t_n\} \in [\![Z]\!] \overset{\text{def}}{\Longleftrightarrow} \\
&\quad \exists h : \{1, \ldots, m\} \to \{1, \ldots, n\} \textit{ injection.} \, (a_{h(i)} = p(i) \textit{ for } i = 1, \ldots, m) \land \\
&\qquad (t_{h(i)} - t_{h(j)} \leq D(i, j) \textit{ for } i \neq j \textit{ and } i, j = 1, \ldots, m) \land \\
&\qquad (-D(0, i) \leq t_{h(i)} \leq D(i, 0) \textit{ for } i = 1, \ldots, m)
\end{aligned}$$

We say that an existential zone Z is consistent *if $[\![Z]\!] \neq \emptyset$.*

Along [13], all we have to show is that we can compute a predecessor *Pre*(Z) of a given existential zone Z as a finite set of existential zones $\{Z_1, \ldots, Z_n\}$ so that $M \to N$ and $N \in [\![Z]\!]$ for some N if and only if $M \in [\![Z_i]\!]$ for some i. For this purpose, we define four operations on existential regions. The first three are from [13], but among them, the addition operation is slightly different from that in [13].

Definition 17. *Suppose $[u : v]$ denotes the interval between u and v. Let $Z = (m, p, D)$ be an existential zone, $i, i_1, \ldots, i_n \in \{1, \ldots, m\}$ be natural numbers, and $a \in S$ be an element. Then the* conjunction $Z \otimes ([u : v], i)$, *the* addition $Z \oplus a$, *the* abstraction $Z \setminus i$, *and* equalization $Z \oslash \{i_1, \ldots, i_n\}$ *are defined as follows:*

conjunction $Z \otimes ([u : v], i) \overset{\text{def}}{=} (m, p, D')$ *where* $D'(i, 0) = \min(v, D(i, 0))$, $D'(0, i) = \min(-u, D(0, i))$, *and* $D(k, j) = D(k, j)$ *for* $k \neq j$, $(k, j) \neq (i, 0)$, *and* $(k, j) \neq (0, i)$.

addition $Z \oplus a \overset{\text{def}}{=} (m + 1, p', D')$ *where* $D'(m + 1, 0) = \infty$, $D'(0, m + 1) = 0$, $D'(m + 1, j) = D'(j, m + 1) = \infty$ *for* $j \in \{1, \ldots, m\}$, $D'(k, j) = D(k, j)$ *for* $j, k \in \{1, \ldots, m\}$, $p'(m + 1) = a$, *and* $p'(j) = p(j)$ *for* $j \in \{1, \ldots, m\}$.

abstraction $Z \setminus i \overset{\text{def}}{=} (m-1, p', D')$ *where* $D'(j, k) = D(j, k)$ *for* $j, k \in \{0, \ldots, i-1\}$, $D'(j, k) = D(j, k + 1)$ *and* $D'(k, j) = D(k + 1, j)$ *for* $j \in \{0, \ldots, i - 1\}$ *and* $k \in \{i, \ldots, m - 1\}$, $D'(j, k) = D(j + 1, k + 1)$ *for* $j, k \in \{i, \ldots, m - 1\}$, $p'(j) = p(j)$ *for* $j \in \{0, \ldots, i-1\}$, *and* $p'(j) = p(j+1)$ *for* $j \in \{i, \ldots, m-1\}$.

equalization $Z \oslash \{i_1, \ldots, i_n\} \overset{\text{def}}{=} (m, p, D')$ *where* $D'(j, k) = 0$ *if* $j, k \in \{i_1, \ldots, i_n\}$, *otherwise* $D'(j, k) = D(j, k)$.

Lemma 8. *For a TMSRS $\mathcal{R}$, and a jump rule*

$$R : a_1{:}t_1, \ldots, a_k{:}t_k \to b_1{:}s_1, \ldots, b_l{:}s_l \textbf{ if } \mathcal{C},$$

the predecessor $Pre_R(Z)$ of a given existential zone $Z = (m, p, D)$ with respect to R is given by the smallest set containing each existential zone Z' such that there is a partial injection $h : \{1, \ldots, m\} \to \{1, \ldots, l\}$ with a domain $\{i_1, \ldots, i_n\}$, and existential zones Z_1, Z_2, and Z_3 satisfying the following conditions:

- $p(i_j) = b_{h(i_j)}$ *for* $j \in \{1, \ldots, n\}$.
- $Z_1 = Z \oplus a_1 \oplus \cdots \oplus a_k$.
- $Z_2 = Z_1 \otimes (\mathcal{C}(s_{h(i_1)}), i_1) \otimes \cdots \otimes (\mathcal{C}(s_{h(i_n)}), i_n) \otimes (\mathcal{C}(t_1), m+1) \otimes \cdots \otimes (\mathcal{C}(t_k), m+k)$.
- $Z_3 = Z_2 \oslash K(x_1) \oslash \cdots \oslash K(x_r)$ *is consistent.*
- $Z' = \widetilde{Z_3} \setminus i_1 \setminus \cdots \setminus i_n$

where

- $\mathcal{C}(t) \overset{\text{def}}{=} [t : t]$ *if* $t \in \mathbb{N}$, *otherwise* $\mathcal{C}(t) \overset{\text{def}}{=} [u : v]$ *satisfying* $\mathcal{C} \Leftrightarrow (t \in [u : v]) \wedge \mathcal{C}'$ *for some* $\mathcal{C}' \in \mathcal{B}(\mathrm{Var}(\mathcal{C}) \setminus \{t\})$, *where* $\mathrm{Var}(\mathcal{C})$ *is the set of clock variables occurring in* $\mathcal{C}$.
- $\{x_1, \ldots, x_r\} = \mathrm{Var}(\mathcal{C})$ *and* $K(x_i) \overset{\text{def}}{=} \{i_j \mid x_i = s_{h(i_j)}, j \in \{1, \ldots, n\}\} \cup \{m + j \mid x_i = t_j, j \in \{1, \ldots, k\}\}$.

– $\widetilde{Z_3} = (m_3, p_3, \widetilde{D_3})$ *is a "normal form" of* Z_3*, i.e.,* $[\![\widetilde{Z_3}]\!] = [\![Z_3]\!]$ *and* $\widetilde{D_3}(i,j) \leq \widetilde{D_3}(i,q) + \widetilde{D_3}(q,j)$ *for* $i, j, q \in \{0, \ldots, m+k\}$*.*

Together with the lemma that states the predecessor of $\xrightarrow{\mathcal{T}}$ is computable in [13], we have that $Pre(Z)$ is computable for any existential zone Z.

Theorem 11. *The coverability of rewriting on existential zones is decidable. Therefore the coverability of a TMSRS from the timed multiset whose clocks are natural numbers is also decidable.*

5 With Diagonal Constraints

For timed automata, it is known that the expressiveness is not increased even if we allow diagonal constraints ($x - y \bowtie c$) as transition conditions [9]. Moreover, if clocks are updatable by constants ($x := c$) or by copying ($x := y$) through transition through edges, the expressiveness remains the same. There are some classes of updatable timed automata [18] such that the emptiness problem is undecidable if the automaton has diagonal constraints whereas it is decidable if the automaton is diagonal-free.

Here we consider how diagonal constraints affect the expressiveness of TMSRSs. Note that we already have the implicit restricted form of diagonal constraints such that $x - y = 0$ since we can use the same clock variable more than once in a single jump rule. Since the three properties on a general TMSRS are already undecidable, we consider the TMSRSs with diagonal constraints without invariant rules. Contrary to the case of timed automata, all three properties become undecidable once a TMSRS contain diagonal constraints even if clocks are updated only by constants or copying.

Definition 18. *We define a simulation of 2-counter machines by a TMSRS with diagonal constraints. The value* n_i *of the counter* r_i *is represented by a multiset* $\{a_i{:}0, \ldots, a_i{:}n_i{-}1, b_i{:}n_i\}$*.*

– *Increment:* $I(i,j,k)$

$$I_j^1 : p_j{:}1,\ a_{3-i}{:}t,\ b_{3-i}{:}s \rightarrow p_k{:}0,\ a_i{:}0,\ a_{3-i}{:}0,\ b_{3-i}{:}t\ \textbf{\textit{if}}\ s - t = 1$$
$$I_j^2 : p_j{:}1,\ b_{3-i}{:}1 \rightarrow p_k{:}0,\ a_i{:}0,\ b_{3-i}{:}0$$

– *Test&Decrement:* $D(i,j,k,l)$

$$I_j^1 : p_j{:}0,\ a_i{:}t,\ b_i{:}s \rightarrow p_k{:}0,\ b_i{:}t\ \textbf{\textit{if}}\ s - t = 1 \qquad I_j^2 : p_j{:}0,\ b_i{:}0 \rightarrow p_l{:}0,\ b_i{:}0$$

The simulation of Test&Decrement is straightforward, and there is no nondeterminism as in Definition 8. To increment a counter r_i, we first wait 1 time unit, then add a timed element $a_i{:}0$ and 'rewind' the clock of the other counter r_{3-i} by replacing $\{a_{3-i}{:}t,\ b_{3-i}{:}t{+}1\}$ with $\{a_{3-i}{:}0,\ b_{3-i}{:}t\}$ (or $\{b_{3-i}{:}1\}$ with $\{b_{3-i}{:}0\}$).

The above definition enables us to simulate an execution in a 2-counter machine by a TMSRS, and the undecidability of reachability, boundedness, and coverability can be shown with an argument similar to Section 3.2.

6 Conclusion

In this paper, we discussed the decidability of reachability, boundedness, and coverability of TMSRSs with or without invariant rules and diagonal constraints. All the three properties for TMSRSs with invariant rules, and the reachability of TMSRSs without invariant rules are undecidable, and they are shown by simulating 2-counter machines whose halting problem is undecidable. The boundedness and coverability of TMSRSs without invariant rules are decidable, and they are shown using regions, Karp-Miller trees, and existential zones. If we allow diagonal constraints in the rules, then all three properties are undecidable even if we disallow invariant rules.

The results of undecidability with invariant rules are too coarse to apply to the simulation of timed automata. We anticipate that if the elements in invariant rules are bounded, which is true for the simulation of timed automata, then boundedness and coverability would be decidable. Including this aspect, we should explore the system more in detail in order to make fine distinction between decidable classes.

Acknowledgement

We are grateful to Koichi Takahashi from National Institute of Advanced Industrial Science and Technology for commenting the early versions of the paper. We also thank anonymous referees for constructive and informative comments.

This work was partially supported by the Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Exploratory Research, 14658088, 2002, Grant-in-Aid for Scientific Research on Priority Areas (B), 12133101, 2001, and Grant-in-Aid for Scientific Research on Priority Areas (C), 13224012, 2001.

References

1. Meseguer, J.: Conditional rewriting logic as a unified model of concurrency. Theoretical Computer Science **96** (1992) 73–155
2. Cervesato, I., Durgin, N.A., Lincoln, P., Mitchell, J.C., Scedrov, A.: A meta-notation for protocol analysis. In: IEEE CSFW. (1999) 55–69
3. Kosiuczenko, P., Wirsing, M.: Timed rewriting logic for the specification of time-sensitive systems. In Schwichtenberg, H., ed.: Proceedings of the Internat. Summer School on Proof and Computation. (1995)
4. Ölveczky, P.C., Meseguer, J.: Specifying real-time systems in rewriting logic. In: Electronic Notes in Theoretical Computer Science. Volume 4. (1996)
5. Kanovich, M., Okada, M., Scedrov, A.: Specifying real-time finite-state systems in linear logic (1998)
6. Hagiya, M., Yamamoto, M., Cottin, J.M.: Symbolic analysis of timed multiset rewriting and its application to protocol analysis (extended abstract). In: Rewriting in Proof and Computation, International Workshop, RPC'01, The Research Institute of Electrical Communication (RIEC), Tohoku University (2001) 34–41

7. Bozzano, M., Delzanno, G., Martelli, M.: An effective bottom-up semantics for first-order linear logic programs. In: FLOPS. (2001) 138–152
8. Cerone, A., Maggiolo-Schettini, A.: Time-based expressivity of time Petri nets for system specification. Theoretical Computer Science **216** (1999) 1–53
9. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science **126** (1994) 183–236
10. Henzinger, T.: The theory of hybrid automata. In: Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS '96), New Brunswick, New Jersey (1996) 278–292
11. Ruiz, V.V., Gomez, F.C., de Frutos Escrig, D.: On non-decidability of reachability for timed-arc Petri nets. In: Proc. 8th Int. Workshop on Petri Net and Performance Models (PNPM'99), 8-10 October 1999, Zaragoza, Spain. (1999) 188–196
12. de Frutos Escrig, D., Ruiz, V.V., Alonso, O.M.: Decidability of properties of timed-arc Petri nets. In Nielsen, M., Simpson, D., eds.: Lecture Notes in Computer Science: 21st International Conference on Application and Theory of Petri Nets (ICATPN 2000), Aarhus, Denmark, June 2000. Volume 1825., Springer-Verlag (2000) 187–206
13. Abdulla, P.A., Nylén, A.: Timed Petri nets and BQOs. In: Proc. ICATPN'2001, 22nd Int. Conf. on application and theory of Petri nets. Volume 2075 of LNCS. (2001) 53–70
14. T.A. Henzinger, X. Nicollin, J. Sifakis, S. Yovine: Symbolic Model Checking for Real-Time Systems. In: 7th. Symposium of Logics in Computer Science, Santa-Cruz, California, IEEE Computer Scienty Press (1992) 394–406
15. Alur, R., Henzinger, T.A.: Real-time system = discrete system + clock variables. International Journal on Software Tools for Technology Transfer **1** (1997) 86–109
16. Abdulla, P.A., Jonsson, B.: Verifying networks of timed processes. Lecture Notes in Computer Science **1384** (1998) 298–312
17. Finkel, A., Schnoebelen, P.: Well-structured transition systems everywhere ! Theoretical Computer Science **256** (2001) 64–92
18. Bouyer, P., Dufourd, C., Fleury, E., Petit, A.: Are timed automata updatable ? In: Proc. 12th Int. Conf. Computer Aided Verification (CAV'2000), Chicago, IL, USA, July 2000. Volume 1855., Springer (2000) 464–479