Book Title
Book Editors
IOS Press, 2003

A class of automata for computing reachability relations in timed systems

Cătălin Dima

Université Paris 12 – Val de Marne, 61 av. du Général de Gaulle, 94010 Créteil Cedex

Abstract. We give an algorithmic calculus of the reachability relations on clock values defined by timed automata. Our approach is a modular one, by computing unions, compositions and reflexive-transitive closure (star) of "atomic" relations. The essential tool is a new representation technique for n-clock relations – the $2n$-automata – and our strategy is to show the closure under union, composition and star of the class of $2n$-automata that represent reachability relations in timed automata.

Keywords. Timed automata, Difference bound matrices.

1. Introduction

Timed automata [1] are a successful and widely used extension of finite automata for modeling real-time systems [17,13]. They are finite automata augmented with clocks that measure time passage. Clocks evolve synchronously at rate 1, and may have any real value. Transitions are taken when some simple arithmetic conditions on the clocks are met, and some transitions might reset some clocks to 0.

Several problems in the domain of verification of real-time systems translate to checking whether a set of states is reachable in a timed automaton. The possibility to reach a state r, starting from a state q, depends essentially on the clock values with which we start in state q. Moreover, even if r is reachable by starting from q with a given vector v of clock values, the set of clock values when reaching r is in general uncountable. The dependency between clock values when starting from q and clock values when reaching r, that is, the *reachability relation* defined by q and r, is therefore an important characteristics for the timed automaton. Its computation might help in the verification of some clock constraints which cannot be modeled with timed automata transitions [11]. The symbolic computation of such relations is the subject of this paper.

Our approach is to build the reachability relations by computing unions, compositions and reflexive-transitive closures of "atomic" relations. To this end, we introduce $2n$-*automata* as a new representation technique for n-clock relations. The idea is that each tuple of clock values in the given relation is coded by a *run* in the automaton. We then show that these automata are closed under union, composition and (for mild conditions) star and have a decidable emptiness problem. We also show that the mild conditions are necessary since in general star closure is not possible, and that the $2n$-automata arising from the relational semantics of timed automata satisfy these conditions.

The representation of clock relations by $2n$-automata passes through several intermediary steps in which we make extensive use of the *difference bound matrix* (DBM) representation [3] of "diagonal" constraints (i.e., constraints like $x - y \in]2, 5]$). In fact, our $2n$-automata represent *unions of DBMs*, and therefore give a new technique for consistency checking of disjunctions of diagonal constraints [14].

Expressing reachability relations in timed automata by means of Presburger Arithmetic is the subject of [5]. The technique employed in that paper is to construct the reflexive-transitive closure of the constraint graph associated to simple loops in timed automata. The problem is that constraint graphs cannot represent constraints employing disjunctions, whereas the reflexive-transitive closure is an infinite disjunction. Therefore, in [5] the authors need to "flatten" each timed automaton, such that no nested loops be allowed, and then "accelerate" each simple loop. On the contrary, our $2n$-automata can represent disjunctions and therefore we may iterate the reflexive-transitive closure construction. In other words, our construction allows the iteration of "sets of loops", not only of simple loops. Another result on expressing clock relations in pushdown timed automata, which utilizes a technique for representing dense clock constraints with discrete clocks, similar to our representation technique, can be found in [6].

This paper complements [8], in that it gives a full proof of the star-closure theorem for n-automata. The proof is also different from the one that can be found in [7].

The paper runs as follows: in the second section we remind the definition of timed automata (without actions) and give a relational semantics for them. In the third section we give several properties of DBMs and investigate on the possibility to model relational composition and reflexive-transitive closure by DBM relations. We emphasize several problems that such a modeling cause, and our solution is to decompose DBMs into *region matrices* and to utilize set-based operations on region matrices. The operations on region matrices are introduced and studied in the fourth section. The fifth section contains the definition and properties of n-automata This section is devoted only to representing sets of *point* region matrices. The sixth section provides the central result of this paper, our star-closure theorem for $2n$-automata whose language satisfy a so-called *non-elasticity* property. We give here a simpler proof, compared to [7]. In the seventh section we show how to represent sets of *nonpoint* region matrices too – and hence DBMs – with n-automata. We also show here that the $2n$-automata arising from the relational semantics of timed automata satisfy the non-elasticity condition that assures star closure.

2. Timed automata

In this section we remind the definition of timed automata and their timed transition semantics [1]. We then show how to transform this semantics into a relational semantics, by associating to each transition a relation on the clock values before, resp. after taking the transition. This relational semantics uses *reset points* instead of clock values, for reasons to be explained later in the paper. Let us note that, as we are not interested in computing *behaviors* of timed automata, we do not endow them with actions or state labels.

First, we give several notations: the set of real nonnegative numbers is denoted $\mathbb{R}_{\geq 0}$ while the sets of integers, resp. nonnegative integers, are denoted $\mathbb{Z}$, resp. $\mathbb{N}$. For any finite set A, $card(A)$ denotes the cardinality of A. The term *interval* denotes any interval whose extremities are integers, or ∞ or $-\infty$ and whose parentheses are either open or closed. For example, $[-1,\infty[= \{x \in \mathbb{R} \mid -1 \leq x < \infty\}$. The empty set is an interval too, and is represented in a unique way – that is, we do not allow representations like $[3, 2[$ for $\emptyset$.

We use the set-lifting of summation of reals, that is, for $A, B \subseteq \mathbb{R}$, $A + B = \{a + b \mid a, b \in \mathbb{R}\}$. Note that for A, B intervals, $A + B$ is an interval whose limits and parentheses can be easily computed from A and B. E.g., $[2, 3[+ [1, 4] = [3, 7[$. We also have $I + \emptyset = \emptyset + I = \emptyset$, for any interval I.

A timed automaton is a finite automaton [10] endowed with the possibility to measure time passage by means of real-valued clocks. The automaton has a finite set of *states* and a finite set of *transitions*, which are "commands" for state changing. The automaton may rest in some state for a finite amount of time t, during which the clocks values are incremented with t. Then it may "move" to another state by taking one of the "enabled" transitions. For a transition to be enabled, the clock values must satisfy a certain constraint. Taking a transition has two effects: the state changes and some clocks are reset to zero.

Formally, a **timed automaton** with clocks $\{x_1, \ldots, x_n\}$ is a tuple $\mathcal{A} = (Q, \delta)$ where Q is a finite set of *states* and δ is a finite set of tuples (*transitions*) (q, C, X, q') where $q, q' \in Q$, $X \subseteq \{1, \ldots, n\}$ and C is a conjunction of *clock constraints* of the following two forms:

1. $x_i \in I$, where $i \in \{1, n, \ldots,\}$ and $I \subseteq [0, \infty[$ is an interval.
2. $x_i - x_j \in J$, where $i, j \in \{1, \ldots, n\}$, $i \neq j$ and $J \subseteq]-\infty, \infty[$ is an interval. We will call such two-variable constraints *diagonal constraints*.

Each timed automaton can be represented as a labeled graph, with states represented as nodes and transitions as labeled edges. An example is provided in Figure 1.

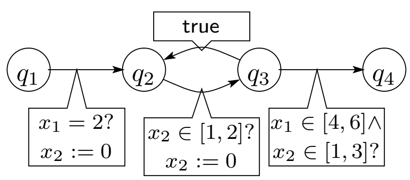

Figure 1. A timed automaton.

In this paper we will focus on an alternative semantics of timed automata: the *reset points semantics*, first used in [4]. The idea is to utilize a new clock T which is never reset, and to record, at each moment t and for each clock x_i, the last moment before t when x_i was reset – denote this by r_i. Then the value of the clock x_i at t equals $T - r_i$. Furthermore, time passage in a state amounts to incrementing T, while resetting clock x_i amounts to the assignment $r_i := T$.

Formally, the *reset transition system* associated to $\mathcal{A}$ is $\mathcal{T}(\mathcal{A}) = (Q \times \mathbb{R}^{n+1}_{\geq 0}, \theta)$ where $\mathbb{R}^{n+1}_{\geq 0}$ is the set of $(n+1)$-tuples of nonnegative reals and $\theta \subseteq Q \times \mathbb{R}^{n+1}_{\geq 0} \times Q \times \mathbb{R}^{n+1}_{\geq 0}$ is defined as follows:

$$\theta = \big\{(q, r_1, \ldots, r_n, t, q, r_1, \ldots, r_n, t') \mid t' \geq t, r_i \leq t \;\forall 1 \leq i \leq n\big\}$$
$$\cup \big\{(q, r_1, \ldots, r_n, t, q', r'_1, \ldots, r'_n, t) \mid \exists (q, C, X, q') \in \delta \text{ s.t. } \forall 1 \leq i \leq n, r_i \leq t \text{ and}$$
$$i \in X \Rightarrow r'_i = t', i \notin X \Rightarrow r'_i = r_i \text{ and for } v_i := t' - r_i, (v_1, \ldots, v_n) \models C\big\}.$$

A *run* in $\mathcal{T}(\mathcal{A})$ is a sequence of configurations connected by transitions in θ. The set of runs of $\mathcal{T}(\mathcal{A})$ gives the (reset points) *semantics* of $\mathcal{A}$. For our example in Figure 1, the following is a run in $\mathcal{T}(\mathcal{A})$: $(q_1, 0, 0, 1) \to (q_2, 0, 0, 2) \to (q_2, 0, 2, 2) \to (q_2, 0, 2, 3.5) \to (q_3, 0, 3.5, 3.5) \to (q_3, 0, 3.5, 5) \to (q_4, 0, 3.5, 5)$.

The relations that we want to compute in this paper are all the possible dependencies between initial and final reset values in a run that starts in some given state q and ends in a given state q', i.e., the relations $R_{qq'} \subseteq \mathbb{R}^{2n+2}_{\geq 0}$ defined by

$$R_{qq'} = \big\{(r_1, \ldots, r_n, t, r'_1, \ldots, r'_n, t') \mid \exists \text{ a run } \big(q^j, r^j_1, \ldots, r^j_n, t^j\big)_{j \in \{1, \ldots, k\}} \text{ in } \mathcal{T}(\mathcal{A})$$
$$\text{such that } q^1 = q, q^k = q', t^1 = t, t^k = t' \text{ and } \forall i \in \{1, \ldots, n\}, r^1_i = r_i, r^k_i = r'_i\big\}.$$

Our aim is to give a characterization of the relations $R_{qq'}$ which can be computed in a modular way – that is, as unions, compositions and/or reflexive-transitive closure of some elementary relations – and which must have a decidable emptiness problem. Here, by composition we mean the following operation: $R \circ R' = \big\{(a, a'') \mid \exists a' \in A \text{ such that } (a, a') \in R \text{ and } (a', a'') \in R'\big\}$, while the reflexive-transitive closure of a relation R is $R^* = \bigcup_{n \geq 1} \underbrace{R \circ \ldots \circ R}_{n \text{ times}} \cup \big\{(a, a) \mid a \in A\big\}$.

It is easy to observe how each relation $R_{qq'}$ in a timed automaton may be represented in a modular way: first, we associate to each transition an "atomic" relation on reset points by "fusing" a time passage step with a state change step. Then, we recursively compute unions, compositions and reflexive-transitive closure of these atomic relations by applying the Kleene theorem [10]. For our example in Figure 1, the relation associated to the transition $\tau_{12} = (q_1, x_1 = 2, \{x_2\}, q_2)$ is

$$R_{\tau_{12}} = \big\{(r_1, r_2, t, r'_1, r'_2, t') \mid r_1, r_2, r'_1, r'_2, t, t' \in \mathbb{R}_{\geq 0},$$
$$t' \geq t, r_1 \leq t, r_2 \leq t, t' - r_1 = 2, r'_1 = r_1, r'_2 = t'\big\}.$$

Observe that $(r_1, r_2, t, r'_1, r'_2, t') \in R_{\tau_{12}}$ iff in $\mathcal{T}(\mathcal{A})$ we have the run $(q_1, r_1, r_2, t) \to (q_1, r_1, r_2, t') \to (q_2, r'_1, r'_2, t')$.

Further in this example, if we denote by τ_{ij} the transition connecting state q_i with state q_j then $R_{q_1 q_4} = R_{\tau_{12}} \circ \big(R_{\tau_{23}} \circ R_{\tau_{32}}\big)^* \circ R_{\tau_{23}} \circ R_{\tau_{34}}$.

In general, given a transition $\tau = (q, C, X, q')$, with $C = \bigwedge_{i \in \mathcal{I}} x_i \in U_i \wedge \bigwedge_{(i,j) \in \mathcal{J}} x_i - x_j \in U_{ij}$ for some $\mathcal{I} \subseteq \{1, \ldots, n\}$, $\mathcal{J} \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$, we associate the atomic relation $R_\tau \subseteq \mathbb{R}^{2n+2}_{\geq 0}$

$$R_\tau = \big\{(r_1 \ldots r_n, t, r'_1 \ldots r'_n, t') \mid t' \geq t \text{ and } \forall i \in \{1, \ldots, n\}, r_i \leq t, \text{ if } i \in X \text{ then } r'_i = t',$$
$$\text{if } i \notin X \text{ then } r'_i = r_i, \text{ and if we put } v_i = t' - r_i \text{ then } (v_1, \ldots, v_n) \models C\big\}.$$

This relation can be specified by the following formula (which can be rewritten as a formula of linear arithmetic over the reals):

$$f_\tau : x'_{n+1} \geq x_{n+1} \wedge \bigwedge_{i\in\mathcal{I}} x'_{n+1} - x_i \in U_i \wedge \bigwedge_{(i,j)\in\mathcal{J}} x_i - x_j \in U_{ij} \wedge$$
$$\bigwedge_{i\in X} x'_i = x'_{n+1} \wedge \bigwedge_{i\notin X} x'_i = x_i \wedge \bigwedge_{i\in\{1,\dots,n\}} x_i \leq x_{n+1}. \quad (1)$$

Note that this formula uses only diagonal constraints.

Relational composition can be specified by using conjunction and quantification: given two transitions τ_1 and τ_2, specified by the formulas f_{τ_1}, resp. f_{τ_2}, the composition $R_{\tau_1} \circ R_{\tau_2}$ is specified by the formula

$$\exists x'_1 \dots \exists x'_{n+1}. f_{\tau_1}(x_1, \dots, x_{n+1}, x'_1, \dots, x'_{n+1}) \wedge f_{\tau_2}(x'_1 \dots, x'_{n+1}, x''_1, \dots, x''_{n+1}).$$

In the resulting formula we may eliminate the existential quantifiers by arithmetic manipulations. The essential observation is that we will then obtain a *conjunction of diagonal constraints*! This would not be the case if we used the traditional clock valuation semantics – see [5,7].

One of the problems of this approach is that the reflexive-transitive closure of a relation is not expressible in first-order logic [16]. We will therefore use a different formalism: we will code the formulas f_τ by *difference bound matrices* (DBMs) [3]. Our technique is to build a Kleene algebra [12] over the set of DBMs. The rest of the paper is dedicated to constructing operations on DBMs that model composition and reflexive-transitive closure. Remind that quantifier elimination in the formulas specifying compositions of clock relations yields conjunctions of diagonal constraints – this is the source of our idea of defining a composition operation on DBMs.

3. Difference bound matrices

Throughout this section we review some known facts about DBMs and emphasize some hurdles on the way to define a composition operation on DBMs. The solution to these problems is the subject of the next section.

Definition 1 *An n-**DBM** is an $n\times n$ matrix of intervals. The set of n-DBMs is denoted* Dbm_n.

*An n-DBM D is said to be **in normal form** iff for each $i, j, k \in \{1, \dots, n\}$, $D_{ii} = \{0\}$ and $D_{ik} \subseteq D_{ij} + D_{jk}$.*

DBMs represent sets of tuples of real numbers. The *semantics* of n-DBMs is the mapping $\|\cdot\| : \mathsf{Dbm}_n \to \mathbb{R}^n$ defined as follows: for all $D \in \mathsf{Dbm}_n$,

$$\|D\| \stackrel{def}{=} \big\{(a_1, \dots, a_n)\in\mathbb{R}^n \mid \forall i, j \in\{1, \dots, n\}, a_j - a_i \in D_{ij}\big\}.$$

Observe that, in an n-DBM D in normal form, $D_{ij} \neq \emptyset$ for all $i, j \in \{1, \dots, n\}$.

Checking DBMs for emptiness is achieved with the following property:

Proposition 2 ([9])
1. *For each $D \in \mathsf{Dbm}_n$, if $\|D\| \neq \emptyset$ then for each sequence of indices $(i_j)_{j\in\{1,\dots,k+1\}}$ with $i_j \in \{1, \dots, n\}$ and $i_{k+1} = i_1$ we have that $0 \in \sum_{j=1}^{k} D_{i_j i_{j+1}}$.*
2. *For each DBM D with the above property, there exists a* unique *DBM in normal form D' with $\|D\| = \|D'\|$. Moreover, D' can be effectively computed from D.*

3. *Any n-DBM in normal form has a nonempty semantics.*

Note also that an n-DBM in normal form is minimal w.r.t. the partial order '$\sqsubseteq$' $\subseteq \mathsf{Dbm}_n \times \mathsf{Dbm}_n$, defined by $D \sqsubseteq D'$ iff $D_{ij} \subseteq D'_{ij}$ for all $i, j \in \{1, \dots, n\}$.

We want to define an operation on DBMs that simulates relational composition. Remember that, for formulas f_τ corresponding to a transition τ in a timed automaton, composition was implemented as conjunction followed by quantification. We therefore define an operation on DBMs which models conjunction, and an operation which models quantification. The first will be called *juxtaposition*, and the second *projection*, for reasons that will come up in their definition. We define these operations at the semantic level first, that is, on tuples of reals:

Definition 3 *Given an n-tuple $\overline{a} = (a_1, \dots, a_n) \in \mathbb{R}^n_{\geq 0}$ and a set $X \subseteq \{1, \dots, n\}$, with $X = \{i_1, \dots, i_k\}$ where $i_j < i_{j+1}$ for all $j \in \{1, \dots, k-1\}$, the X**-projection** of $\overline{a}$ is*

$$\overline{a}\big|_X \stackrel{def}{=} (a_{i_1}, \dots, a_{i_k}) \in \mathbb{R}^{card(X)}_{\geq 0}.$$

*Given $m, n, p \in \mathbb{N}$ with $p \leq \min(m, n)$, the p**-juxtaposition** of $\overline{a} = (a_1, \dots, a_m) \in \mathbb{R}^m_{\geq 0}$ with $\overline{b} = (b_1, \dots, b_n) \in \mathbb{R}^n_{\geq 0}$ is defined iff $a_{m-p+i} = b_i$ for all $i \in \{1, \dots, p\}$ and is the $(m+n-p)$-tuple denoted $\overline{a} \,\square_p\, \overline{b}$, whose i-th component is:*

$$\left(\overline{a} \,\square_p\, \overline{b}\right)_i \stackrel{def}{=} \begin{cases} a_i & \text{for } i \in \{1, \dots, m\} \\ b_{i-m+p} & \text{for } i \in \{m+1, \dots, m+n-p\}. \end{cases}$$

*Given $\overline{a}, \overline{b} \in \mathbb{R}^{2n}_{\geq 0}$, the **composition** of $\overline{a}$ with $\overline{b}$ is the tuple denoted $\overline{a} \odot \overline{b}$ and obtained as follows:*

$$\overline{a} \odot \overline{b} \stackrel{def}{=} \left(\overline{a} \,\square_n\, \overline{b}\right)\big|_{\{1,\dots,n\} \cup \{2n+1,\dots,3n\}} \in \mathbb{R}^{2n}_{\geq 0}.$$

It is trivial to check that juxtaposition is associative, that is, $(\overline{a} \,\square_q\, \overline{b}) \,\square_r\, \overline{c}$ is defined iff so is $\overline{a} \,\square_q (\overline{b} \,\square_r\, \overline{c})$, and then $(\overline{a} \,\square_q\, \overline{b}) \,\square_r\, \overline{c} = \overline{a} \,\square_q (\overline{b} \,\square_r\, \overline{c})$. *Composition* is then associative too but does not have a unit.

The *powerset* of $2n$-tuples can then be endowed with an associative composition by lifting $\odot$ to sets: for $A, B \subseteq \mathbb{R}^{2n}_{\geq 0}$, $A \odot B = \left\{\overline{a} \odot \overline{b} \mid \overline{a} \in A, \overline{b} \in B\right\}$. The unit for *set composition* is the set

$$\mathbf{1}_{2n} \stackrel{def}{=} \left\{(\overline{a}, \overline{a}) \mid \overline{a} \in \mathbb{R}^n_{\geq 0}\right\} = \left\{(a_1, \dots, a_n, a_1, \dots, a_n) \mid a_i \in \mathbb{R}_{\geq 0} \forall i \in \{1, \dots, n\}\right\}.$$

Moreover, for each $L \subseteq \mathbb{R}^{2n}_{\geq 0}$, we may define the *star* of L as the set $L^{\circledast} = \bigcup_{k \in \mathbb{N}} L^{k\odot}$, where $L^{0\odot} = \mathbf{1}_{2n}$ and for all $k \in \mathbb{N}$, $L^{(k+1)\odot} = L^{k\odot} \odot L$.

Proposition 4 $\left(\mathcal{P}(\mathbb{R}^{2n}_{\geq 0}), \cup, \odot, \circledast, \emptyset, \mathbf{1}_{2n}\right)$ *is a Kleene algebra (in the sense of [12]).*

Our aim is to lift these operations to DBMs in a *compositional* manner – that is, to define operations $\big|_X$, $\square_p$, $\odot$ and $\circledast$ on DBMs such that

$$\|D_1 \,\square_p\, D_2\| = \|D_1\| \,\square_p\, \|D_2\|, \qquad \|D\big|_X\| = \|D\|\big|_X$$
$$\|D_1 \odot D_2\| = \|D_1\| \odot \|D_2\|, \qquad \|D^{\circledast}\| = \|D\|^{\circledast}$$

Projection. The X-projection of an n-DBM D is, intuitively, the removal from D of the rows and columns whose indices are not in X:

Definition 5 *Given $D \in \mathsf{Dbm}_n$ and a set $X \subseteq \{1,\dots,n\}$ with $X = \{k_1,\dots,k_m\}$, where $k_i < k_{i+1}$ for all $i \in \{1,\dots,m-1\}$, the X**-projection** of D is the $card(X)$-DBM $D\big|_X$ whose (i,j)-component is:*

$$\big(D\big|_X\big)_{ij} \stackrel{def}{=} D_{k_i k_j}.$$

Note that, on arbitrary DBMs, projection is *not compositional*: for example, the 3-DBM $D = \begin{pmatrix} \{0\} & \{1\} & \{1\} \\ \{-1\} & 0 & \{1\} \\ \{-1\} & \{-1\} & 0 \end{pmatrix}$ has an empty semantics, but its projection on the set $\{1,2\}$ has a nonempty semantics, since $(0,1) \in \|D\|\big|_{\{1,2\}}$.

Proposition 6 $\|D\big|_X\| = \|D\|\big|_X$*, for any $D \in \mathsf{Dbm}_n$ and $X \subseteq \{1,\dots,n\}$.*

Hence, as a corollary of Proposition 2, projection is *compositional* on n-DBMs in normal form.

Juxtaposition. We would now like to define a juxtaposition operation on DBMs, that would verify the equation

$$\|D \,\square_p\, D'\| = \|D\| \,\square_p\, \|D'\|, \tag{2}$$

for all $D \in \mathsf{Dbm}_m$, $D' \in \mathsf{Dbm}_n$ and $p \leq (m,n)$. Several considerations have to be taken into account to this end:

Firstly, note that the set of tuples $\|D\| \,\square_p\, \|D'\|$ is the semantics of a DBM: consider the block-decompositions $D = \begin{pmatrix} D_1 & D_2 \\ D_3 & D_4 \end{pmatrix}$, $D' = \begin{pmatrix} D'_1 & D'_2 \\ D'_3 & D'_4 \end{pmatrix}$, with $D_4, D'_1 \in \mathsf{Dbm}_p$. Then $D \,\square_p\, D' = \begin{pmatrix} D_1 & D_2 & E \\ D_3 & D_4 \cap D'_1 & D'_2 \\ E^t & D'_3 & D'_4 \end{pmatrix}$ where E is a $(m-p) \times (n-p)$ matrix of intervals whose (i,j)-component is $E_{ij} = \bigcap_{k=m-p+1}^{m} D_{ik} + D'_{k-m+p,j+p}$, while E^t is the transpose of E. Observe that, when each component in D and D' is a finite interval, all the components of E are finite intervals too. We might then want to define composition by combining juxtaposition with projection, similar to Definition 3, i.e., $D_1 \odot D_2 \stackrel{def}{=} \big(D_1 \,\square_n\, D_2\big)\big|_{\{1,\dots,n\}\cup\{2n+1,\dots,3n\}}$.

Remind further that we aim at computing reflexive-transitive closure of reachability relations. This amounts to defining $D^{\circledast}$ for each DBM (in normal form, if necessary). This operation needs to be defined compositionally, i.e., it must satisfy the equation

$$\|D^{\circledast}\| = \|D\|^{\circledast} \tag{3}$$

But then this means that $\circledast$ is a *set-based* operation on DBMs in normal form. That is, for each DBM D, it defines the set of DBMs for which the union of their semantics gives $\|D\|^{\circledast}$: take e.g. $D = \begin{pmatrix} \{1\} & \{0\} \\ \{0\} & \{-1\} \end{pmatrix}$. We have that $\|D\|^{\circledast}$ is the union of the semantics of an infinite family of DBMs in normal form, but cannot be represented by a single DBM.

The big problem raised by these definitions is the symbolic calculation of $D^{\circledast}$ – to actually compute it, we need a way to finitely represent $\|D\|^{\circledast}$, and we also need that this finite representation be endowed with an algorithm for emptiness checking.

We start from the observation that Equation 2 can also be satisfied by a *set-based operation*, that is, by an operation $\blacksquare_p$ which associates to each pair (D_1, D_2) a *set* of DBMs whose union of semantics equals $\|D_1\| \blacksquare_p \|D_2\|$. This will be our choice in the next section: we decompose each DBM into the smallest DBMs in normal form, which we call *region matrices*. Then, $D_1 \blacksquare_p D_2$ will be the *set of region matrices* whose semantics is included in $\|D_1\| \blacksquare_p \|D_2\|$. We therefore reduce the problem of representing infinite sets of DBMs to the problem of representing infinite sets of region matrices.

4. Region matrices

This section implements the idea of defining composition on DBMs by decomposing them into region matrices and employing a set-based composition on region matrices. This solution still induces further problems, connected to the fact that we may have to handle *infinite* sets of region matrices. These problems are discussed at the end of this section.

Definition 7 *An n-DBM $D \in \mathsf{Dbm}_n$ is called an n-**region matrix** if it has a nonempty semantics and, for each $i, j \in \{1, \dots, n\}$, there exists some $\alpha \in \mathbb{Z}$ such that either $D_{ij} = \{\alpha\}$ or $D_{ij} =]\alpha, \alpha+1[$.*

Observe that each region matrix is in normal form. The set of region matrices is denoted Reg_n. Their name is drawn from the similarities with the regions of [1]. When a region matrix contains only point intervals, we call it a **point region matrix**.

The following property gives the principle of the decomposition of DBMs into region matrices:

Proposition 8 *For all $D \in \mathsf{Dbm}_n$, $\|D\| = \bigcup \big\{ \|R\| \mid R \in \mathsf{Reg}_n \big\}$.*

Definition 9 *Given two region matrices $R_1 \in \mathsf{Reg}_m$, $R_2 \in \mathsf{Reg}_n$ and $p \le \min(m, n)$, the **region p-juxtaposition** of R_1 and R_2 is the following* set *of $(m+n-p)$-region matrices:*

$$R_1 \blacksquare_p R_2 \stackrel{def}{=} \big\{ R \in \mathsf{Reg}_{m+n-p} \mid R\big|_{\{1,\dots,m\}} = R_1, R\big|_{\{m-p+1,\dots,m+n-p\}} = R_2 \big\}.$$

*Given $R_1, R_2 \in \mathsf{Reg}_{2n}$, the $2n$-**region-composition** of R_1 and R_2 is the set*

$$R_1 \odot R_2 \stackrel{def}{=} \big\{ R\big|_{\{1,\dots,n\} \cup \{2n+1,\dots,3n\}} \mid R \in R_1 \blacksquare_n R_2 \big\}.$$

The composition of two $2n$-DBMs D_1, D_2 is then

$$D_1 \odot D_2 \stackrel{def}{=} \big\{ R \in R_1 \odot R_2 \mid R_i \in \mathsf{Reg}_{2n}, \|R_i\| \subseteq \|D_i\| \textit{ for } i = 1, 2 \big\}.$$

Note that region juxtaposition is associative and compositional: for each $R_1 \in \mathsf{Reg}_m$, $R_2 \in \mathsf{Reg}_n$ and $p \le \min(m, n)$, $\|R_1 \blacksquare_p R_2\| = \|R_1\| \blacksquare_p \|R_2\|$. This also implies that region composition is compositional. Note also that $R_1 \blacksquare_p R_2 = \emptyset$ iff $R_1\big|_{\{m-p+1,\dots,m\}} \neq R_2\big|_{\{1,\dots,p\}}$.

Composition can then be lifted to sets of regions as usual, $\mathcal{R}_1 \odot \mathcal{R}_2 = \big\{ R_1 \odot R_2 \mid R_1 \in \mathcal{R}_1, R_2 \in \mathcal{R}_2 \big\}$. The unit for composition on *sets of region matrices* is the set

$$\mathbf{1}_{2n}^{reg} \stackrel{def}{=} \big\{ R \in \mathsf{Reg}_{2n} \mid \forall i, j \in \{1, \dots, n\}, R_{ij} = R_{n+i,j} = R_{i,n+j} = R_{n+i,n+j} \big\}.$$

The *star* operation $\circledast$ can then be defined as follows: for each $\mathcal{R} \subseteq \mathsf{Reg}_{2n}$, $\mathcal{R}^{\circledast} = \bigcup_{k\in\mathbb{N}} \mathcal{R}^{k\odot}$, where $\mathcal{R}^{0\odot} = \mathbf{1}_{2n}^{reg}$ and $\mathcal{R}^{(k+1)\odot} = \mathcal{R}^{k\odot} \odot \mathcal{R}$ for all $k \in \mathbb{N}$. Of course, this operation can be extended to DBMs using the idea of decomposing a DBM into regions.

Then star is compositional too: $\|\mathcal{R}^{\circledast}\| = \|\mathcal{R}\|^{\circledast}$ for all $\mathcal{R} \subseteq \mathsf{Reg}_{2n}$. Hence, $\big(\mathcal{P}(\mathsf{Reg}_{2n}), \cup, \odot, \circledast, \emptyset, \mathbf{1}_{2n}^{reg}\big)$ is a Kleene algebra.

One of the classic way to handle infinite sets of objects which are endowed with a composition operation is to consider "regular expressions" over these objects. *Regular expressions over DBMs* are then the class of expressions generated by the grammar:

$$E ::= D \mid E \cup E \mid E \odot E \mid E^{\circledast}, \text{ where } D \in \mathsf{Dbm}_{2n}.$$

The semantics of these expressions is straightforward, e.g.: $\|E_1 \odot E_2\| = \|E_1\| \odot \|E_2\|$, $\|E^{\circledast}\| = \|E\|^{\circledast}$, etc.

Theorem 10 *The emptiness problem for regular expressions over $2n$-DBMs is undecidable for $n \geq 2$.*

The proof relies on the possibility to code runs of two-counter machines (see e.g., [10]), as regular expressions over DBMs.

This theorem, however, is not a barrier in our search for a symbolic computation of reachability relations in timed automata: in fact, not all the regular expressions over DBMs arise as semantics of timed automata. We introduce in the next section the concept of n-automata as a means to code "regularity" in sets of region matrices. They will eventually give a subclass of regular expressions over DBMs that is capable of expressing the relational semantics of timed automata and that has a decidable emptiness problem.

5. n-automata

In this section we give the definition and the basic properties of n-automata. In particular, we show that, for n even, $n = 2k$, $2k$-automata are closed under composition. We will identify, in the next section, a special property which assures star closure. To ease the understanding, this section shows how n-automata can be used to represent sets of *point* region matrices. The use of n-automata for representing *nonpoint* region matrices is the subject of Section 7.

Let us denote by PReg_n the set of point n-region matrices. The nice feature of each point n-region matrix is that its semantics contains points in $\mathbb{R}^n_{\geq 0}$ whose coordinates are all natural numbers. And if we represent these numbers in unary, they become words over a one-letter alphabet. This is the intuition behind our definition of n-automata.

Definition 11 *An n-**automaton** is a tuple $\mathcal{A} = (Q, \delta, q_*, Q_1, \dots, Q_n)$ in which Q is a finite set of* states, *$\delta \subseteq Q \times Q$ is a set of* transitions, *and for each $i \in \{1, \dots, n\}$, Q_i is the* accepting component for index *i; its elements are called* accepting states for index *i. Finally, $q_* \in Q$ is called the* initial state *and we assume $(q_*, q_*) \in \delta$ and $\forall i \in \{1, \dots, n\}, \forall q \in Q_i, (q_*, q) \in \delta$.*

An n-automaton accepts a tuple $(a_1, \dots, a_n) \in \mathbb{N}^n$ in the following way: the automaton starts in the specially designated state q_*, and tries to build a run that passes through all the accepting components. For each $i \in \{1, \dots, n\}$, one of the moments when the run passes through Q_i must be exactly when the *length* of the run is a_i. Hence, the transitions of the automaton are time steps and the automaton performs a finite-memory counting of time steps in between the moments when it passes through an accepting state for index i and an accepting state for index j, for each $i, j \in \{1, \dots, n\}$.

More formally, a **run** is a sequence of states $\rho = (q_j)_{j \in \{0,\dots,k\}}$ connected by transitions, i.e., $(q_{j-1}, q_j) \in \delta$ for all $j \in \{1, \dots, n\}$, and with $q_0 = q_*$. The run is **accepting** if for each $i \in \{1, \dots, n\}$ there exists some $j \in \{1, \dots, k\}$ such that $q_j \in Q_i$. Given an n-tuple of naturals $\overline{a} = (a_1, \dots a_n) \in \mathbb{N}^n$ we say that $\overline{a}$ is **accepted by** ρ iff $q_{a_i} \in Q_i$ for all $i \in \{1, \dots, n\}$. The **point language** of $\mathcal{A}$ is then the set of points in $\mathbb{N}^n$ accepted by $\mathcal{A}$, and is denoted $L_p(\mathcal{A})$.

For example, consider the 3-automaton $\mathcal{A}_0$ in Figure 2 below.

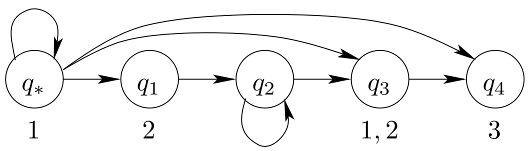

Figure 2. A 3-automaton.

This automaton accepts the 3-tuple $(1, 4, 5)$: the accepting run for it is $q_* \to q_* \to q_1 \to q_2 \to q_3 \to q_4$. Intuitively, we have put all three components of $\overline{a}$ on the real axis and let $\mathcal{A}_0$ "parse" this axis, unit after unit. Note that the tuple $(1, 4, 5)$ belongs to the semantics of the point region matrix $R = \begin{pmatrix} \{0\} & \{3\} & \{4\} \\ \{-3\} & \{0\} & \{1\} \\ \{-4\} & \{-1\} & \{0\} \end{pmatrix}$. We then say that $\mathcal{A}_0$ **accepts** R.

More formally, some $R \in \mathsf{PReg}_n$ is accepted by an n-automaton $\mathcal{A}$ if there exists an n-tuple in $\|R\|$ which is accepted by $\mathcal{A}$. The **region language** of $\mathcal{A}$ is the set of point region matrices accepted by $\mathcal{A}$ and is denoted $L_r(\mathcal{A})$.

Note that the requirement that $(q_*, q_*) \in \delta$ implies that if a tuple $\overline{a} = (a_1, \dots, a_n)$ is accepted by $\mathcal{A}$ then for any $k \in \mathbb{N}$, $\overline{a} + k = (a_1 + k, \dots, a_n + k) \in L_p(\mathcal{A})$. Hence, q_* can be used for looping any number of times at the beginning of any accepting run.

We will also use n-automata with 0-transitions, in which $\delta \subseteq Q \times \{0, 1\} \times Q$. The notions of run, accepting run and point regions accepted by a run are similar to "ordinary" n-automata, except that only the 1-transitions are counted in the acceptance of a tuple. Hence, an accepting run $\rho = (q_{j-1} \xrightarrow{p_j} q_j)_{j \in \{1,\dots,k\}}$ accepts a tuple $\overline{a} = (a_1, \dots a_n) \in \mathbb{N}^n$ if for each $i \in \{1, \dots, n\}$ and index j for which $q_j \in Q_i$ we have that $a_i = \sum_{l \leq j} p_l$ ($p_l \in \{0, 1\}$). It is not very difficult (see [8]) to show that n-automata with 0-transitions have the same expressive power as n-automata without 0-transitions. Note however that 0-transition elimination might involve augmenting the state space with a factor of 2^n.

Proposition 12 *The nonemptiness problem for n-automata is NP-complete.*

The proof of the NP-completeness part of this result can be found in [8,7]. We present here an algorithm for checking emptiness that is an adaptation of the Floyd-Warshall-Kleene algorithm, having thus $O\big(card(Q)^3\big)$ iterations, but in which each iteration might take exponential time. In our algorithm, we associate to each pair $(q, r) \in Q^2$, a set Ξ_{qr} of *subsets* of $\{1, \dots, n\}$, i.e., $\Xi_{qr} \subseteq \mathcal{P}(\{1, \dots, n\})$. The set Ξ_{qr} will have the property that, for each $X \in \Xi_{qr}$ and each $i \in X$, there exists a run from q to r that passes through

Q_i. Once the matrix Ξ is constructed, the answer is "YES" if and only if there exists some pair $(q,r) \in Q^2$ with $\{1,\ldots,n\} \in \Xi_{qr}$.

For the computation of the matrix Ξ, we suppose an ordering of Q is given, say $Q = \{q_1,\ldots,q_p\}$ with $p \in \mathbb{N}$. A special operation on $\mathcal{P}\big(\mathcal{P}(\{1,\ldots,n\})\big)$, denoted $\otimes$, is used. This operation works as follows: given $\mathcal{X},\mathcal{Y} \subseteq \mathcal{P}(\{1,\ldots,n\})$,

$$\mathcal{X} \otimes \mathcal{Y} = \big\{ Z_1 \cup Z_2 \mid Z_1 \in \mathcal{X}, Z_2 \in \mathcal{Y} \big\}$$

The algorithm works by constructing a sequence of matrices $(\Xi_k)_{k\in\{0,\ldots,p\}}$ with

$$(\Xi_0)_{ij} = \begin{cases} \{X\} & \text{iff } i = j \text{ and for all } l \in X, q_i \in Q_l \\ \{\emptyset\} & \text{otherwise} \end{cases}$$

$$(\Xi_{k+1})_{ij} = (\Xi_k)_{i,k+1} \otimes (\Xi_k)_{k+1,k+1} \otimes (\Xi_k)_{k+1,j}$$

Proposition 13 $L_p(\mathcal{A}) \neq \emptyset$ *iff* $\exists i,j \in \{1,\ldots,p\}$ *with* $\{1,\ldots,n\} \in (\Xi_p)_{ij}$.

We may do even better than Proposition 13: we may need to check only a single component in Ξ whether it contains $\{1,\ldots,n\}$. To this end, we pick a new state q_{**}. Similarly to how q_* is used for looping before any run, q_{**} is used for looping *after* any run. We will call q_{**} the *sink state*. We then only have to check whether $\Xi_{q_*,q_{**}}$ contains $\{1,\ldots,n\}$.

Formally, we transform $\mathcal{A}$ into the n-automaton $\tilde{\mathcal{A}} = (\tilde{Q},\tilde{\delta},q_*,Q_1,\ldots,Q_n)$ where:

$$\tilde{Q} = (Q \cup \{q_{**}\}) \text{ and } \tilde{\delta} = \delta \cup \big\{ q_{**} \to q_{**}, q \to q_{**} \mid q \in Q_i, i \in \{1,\ldots,n\} \big\}$$

Definition 14 *The automaton $\tilde{\mathcal{A}}$ is called the* ***completion*** *of $\mathcal{A}$. We also call it a* ***complete*** *n-automaton.*

Proposition 15 ([8])

1. *The class of languages of n-automata is closed under union and intersection.*
2. *Given an n-automaton $\mathcal{A}$ and some index set $J \subseteq \{1,\ldots,n\}$ with $card(J) = m$, $L_p(\mathcal{A})\big|_J$ is accepted by an m-automaton.*
3. *The class of languages of $2n$-automata is closed under composition.*
4. *For each n-DBM $D \in \mathsf{Dbm}_n$, the set $\{R \in \mathsf{PReg}_n \mid R \sqsubseteq D\}$ can be accepted by an n-automaton.*

Note that Theorem 10 implies nonclosure under star for $2n$-automata.

6. Non-elasticity

We give here a property on $L_r(\mathcal{A})$ that assures that $L_r(\mathcal{A})^{\circledast}$ can be accepted by a $2n$-automaton. This property is the following:

For each $i,j \in \{1,\ldots,n\}$, if $a_{n+i} \neq a_i$ and $a_{n+j} \neq a_j$ then $a_{n+i} - a_j \geq 0$ and $a_{n+j} - a_i \geq 0$.

Each $2n$-tuple with this property is called **non-elastic**. When the tuple satisfies the unconditional requirement $a_{n+i} - a_j \geq 0$ and $a_{n+j} - a_i \geq 0$ for all $i,j \in \{1,\ldots,n\}$ we say it is a *strictly non-elastic tuple*.

This notion can be lifted to DBMs in the following way: a $2n$-DBM D is called **non-elastic** if for each $i, j \in \{1, \dots, n\}$, if, whenever $R_{i,n+i} \neq \{0\}$ and $R_{j,n+j} \neq \{0\}$ then $R_{j,n+i} \geq \{0\}$ and $R_{i,n+j} \geq \{0\}$. Here we used the following notation: for two intervals $I, J \subseteq \mathbb{R}$, $I \leq J \iff \forall \alpha \in I, \forall \beta \in J, \alpha \leq \beta$. Note that D is non-elastic iff $\|D\|$ is composed by non-elastic tuples.

The central theorem of this paper is the following:

Theorem 16 *Given a $2n$-automaton $\mathcal{A}$ in which $L_r(\mathcal{A})^{\circledast}$ is composed of non-elastic region matrices only, then $L_r(\mathcal{A})^{\circledast}$ can be recognized by a $2n$-automaton.*

Proof: We will first give the construction for the case of $L_p(\mathcal{A})^{\circledast}$ containing only *strictly* non-elastic tuples. The idea is to use, at each moment, one or two replicas of $\mathcal{A}$, such that, each time one of the replicas has completed an accepting run, the other replica starts building an accepting run. The two replicas work "synchronously": for each $i \in \{1, \dots, n\}$, one of the passages of the first replica through Q_{n+i} must happen exactly when the second replica passes through Q_i. We give the $2n$-automaton for $L_p^{\circledast \geq 2}(\mathcal{A}) := \bigcup_{k \geq 2} L_p(\mathcal{A})^{k\odot}$, which is sufficient, since the $2n$-automaton for $L_p(\mathcal{A})^{\circledast}$ can be obtained by union of this automaton with $\mathcal{A}$ and the $2n$-automaton for $\mathbf{1}_{2n}$.

A graphical presentation of the way a tuple $\overline{a} \in L_p(\mathcal{A})^{\circledast \geq 2}$ must be parsed is given in the following figure. Here, the big strip labeled with $\overline{a}$ shows the zone on the real axis where the components of $\overline{a}$ are located. This tuple is decomposed as $\overline{a} = \overline{a}_1 \odot \dots \odot \overline{a}_k$, with $\overline{a}_i \in L_p(\mathcal{A})$ for all $i \in \{1, \dots, k\}$, and the shadowed strips represent the segments where the components of each $\overline{a}_i$ lie on the real axis.

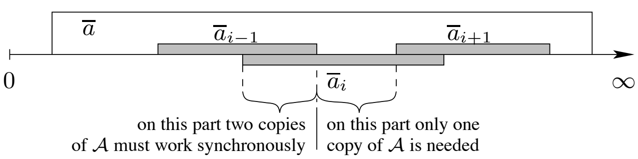

Figure 3. Parsing a strictly non-elastic tuple.

Suppose that $\mathcal{A} = (Q, \delta, q_*, Q_1, \dots, Q_{2n})$. Assume also that $\mathcal{A}$ is complete, that is, there exists the final state q_{**} to which each other state is connected via a transition.

The $2n$-automaton (with 0-transitions!) accepting $L^{\circledast \geq 2}(\mathcal{A})$ is then:

$$\mathcal{B} = \big(Q \cup \big(Q \times Q \times \mathcal{P}(\{1, \dots, n\})\big), \theta, q_*, Q_1, \dots, Q_{2n}\big)$$

$$\begin{aligned}
\theta = \delta \cup \big\{&(q, q', \{1, \dots, n\}) \xrightarrow{0} q' \mid q, q' \in Q\big\} \cup \\
&\big\{q \xrightarrow{0} (q, q', X) \mid X \subseteq \{1, \dots, n\} \text{ and } \forall i \in X, q \in Q_{n+i} \text{ and } q' \in Q_i\big\} \cup \\
&\big\{(q, r, X) \xrightarrow{1} (q', r', Y) \mid q, q', r, r' \in Q, X \subseteq Y \subseteq \{1, \dots, n\} \\
&\qquad \text{and for all } i \in Y \setminus X, q' \in Q_{n+i} \text{ and } r' \in Q_i\big\}
\end{aligned}$$

The structure of an accepting run is the following:

$$\begin{aligned}
\rho = q_* \xrightarrow{1} q_1 \xrightarrow{1} \overset{\dots}{\longrightarrow} \xrightarrow{1} q_{i_1-1} \xrightarrow{0} \big(q_{i_1} = q_{i_1-1}, r_{i_1}, X_{i_1}\big) \xrightarrow{1} \dots \\
\dots \xrightarrow{1} \big(q_{j_1-1}, r_{j_1-1}, X_{j_1-1} = \{1, \dots, n\}\big) \xrightarrow{0} q_{j_1} = r_{j_1-1} \xrightarrow{1} \dots \\
\xrightarrow{1} q_{i_2-1} \xrightarrow{0} \big(q_{i_2} = q_{i_2-1}, r_{i_2}, X_{i_2}\big) \xrightarrow{q}_{j_2} = r_{j_2-1} \xrightarrow{1} \dots \xrightarrow{1} q_k \dots
\end{aligned}$$

Observe that the sequence $\rho_l = \big(q_*, r_{i_l}, r_{i_l+1}, \ldots, r_{j_l-1} = q_{j_l}, q_{j_l+1}, \ldots, q_{j_{l+1}-1}\big)$ is an accepting run in $\mathcal{A}$. This sequence witnesses that the l-th replica of $\mathcal{A}$ works on the l-th segment of the real axis, as suggested in Figure 3 above. Or, on other words, if ρ is associated to some tuple a, then we may decompose the tuple into $a = a_1 \ldots a_k$ such that a_l is associated to ρ_l.

Before going to the general case, let us introduce the following notations: for any $X \subseteq \{1, \ldots, n\}$,

$$\mathrm{Diag}_X = \big\{a \in \mathbb{N}^{2n} \mid a_i = a_{n+i} \text{ for all } i \in X\big\}$$
$$\overline{\mathrm{Diag}}_X = \big\{a \in \mathbb{N}^{2n} \mid a_i \neq a_{n+i} \text{ for all } i \in X\big\}$$

We will denote $\overline{X} := \{1, \ldots, n\} \setminus X$ in the sequel. Note that $\overline{\mathrm{Diag}}_X \neq \mathrm{Diag}_{\overline{X}}$.

Suppose then that there exists some $X \subseteq \{1, \ldots, n\}$ such that $L_p(\mathcal{A}) \subseteq \mathrm{Diag}_X \cap \overline{\mathrm{Diag}}_{\overline{X}}$. In this case, the technique for accepting a tuple $a \in L_(\mathcal{A})^{\circledast\geq 2}$ reuses the technique from the strictly non-elastic case: the parse of the projection $a\big\lfloor_{\overline{X}\cup(\overline{X}+n)}$ is exactly the same as in Figure 3, with the restriction that the replicas are the projections of $\mathcal{A}$ onto $\overline{X} \cup (\overline{X} + n)$.

In a general non-elastic tuple, the components in X may lie anywhere on the axis – with the only constraint that $a_i = a_{n+i}$ for all $i \in X$. We therefore need to start all the replicas at the same time, and to let each one evolve "quietly" outside the interval on the real axis where the $\overline{X}$-components lie.

We will assume here that $\mathcal{A}$ is a *completion*, in the sense of definition 14. That is, each state belonging to an accepting component Q_i has an outgoing transition to the sink state q_{**}. The $2n$-automaton *with* 0-*transitions* accepting $L_(\mathcal{A})^{\circledast\geq 2}$ is $\mathcal{C} = \big(\tilde{\mathcal{Q}}, \tilde{\delta}, \tilde{q}_*, \tilde{Q}_1, \ldots, \tilde{Q}_{2n}\big)$ with $\tilde{Q} = \mathcal{P}(Q) \times Q \times Q \times \mathcal{P}(\overline{X}) \times \mathcal{P}(Q) \times \mathcal{P}(X)$. In a tuple $(S, q, r, Y, T, Z) \in \tilde{Q}$, q is called the **left active component** and plays the role of the first component in the construction of $\mathcal{B}$ above, while r is called the **right active component** and plays the role of the second component in the construction of $\mathcal{B}$ above. Moreover, S is called the **history component** and T the **prophecy component**.

The transitions in $\mathcal{C}$ are as follows:

1. $(S, q, r, Y, T, Z) \xrightarrow{1} (S', q', r', Y', T', Z')$ if and only if
   - $q \xrightarrow{1} q'$ and $r \xrightarrow{1} r'$.
   - $Y \subseteq Y'$ and for all $i \in Y' \setminus Y$, $q' \in Q_{n+i}$ iff $r' \in Q_i$.
   - For all $s \in S$ there exists $s' \in S'$ such that $s \xrightarrow{1} s'$;
   - For all $t' \in T'$ there exists $t \in T$ such that $t \xrightarrow{1} t'$;
   - $Z \subseteq Z'$ and for all $i \in Z' \setminus Z$, $S' \cup T' \cup \{q', r'\} \subseteq Q_i \cap Q_{n+i}$
2. $(S, q, r, \overline{X}, T \cup \{t\}, Z) \xrightarrow{0} (S \cup \{q\}, r, t, Y', T, Z)$ for all $S, T \subseteq \tilde{Q}$, $q, r \in \tilde{Q}$ $Z \subseteq X$ and $Y = \big\{i \in \overline{X} \mid r \in Q_i, t \in Q_i\big\}$.

The initial state for $\mathcal{C}$ is $\tilde{q}_* = \big(\emptyset, q_*, q_*, \emptyset, \{q_*\}, Z\big)$, where $Z = \big\{i \in X \mid q_* \in Q_i\big\}$.

The accepting components for $\mathcal{C}$ are as follows:

1. For each $i \in X$,

$$\tilde{Q}_i = \tilde{Q}_{n+i} = \big\{(S, q, r, Y, T, Z) \mid \exists (S', q, r, Y', T', Z') \in \tilde{Q} \text{ s.t. } i \in Z \setminus Z' \text{ and } (S', q', r', Y', T', Z') \xrightarrow{x} (S, q, r, Y, T, Z),\ \text{with } x = 0 \text{ or } x = 1\big\}$$

2. For each $i \in \overline{X}$,

$$\tilde{Q}_i = \{(\emptyset, q, q', \emptyset, T, Z) \mid q \in Q_i, q' \in Q, T \subseteq Q, Z \subseteq X\}$$
$$\tilde{Q}_{n+i} = \{(S, q, q', \overline{X}, \emptyset, Z) \mid q' \in Q_{n+i}, q \in Q, S \subseteq Q, Z \subseteq X\}$$

The proof of the correctness of this construction follows by double inclusion: consider first an accepting run ρ in $\mathcal{C}$:

$$\rho = \Big(\big(S_{j-1}, q_{j-1}, r_{j-1}, Y_{j-1}, T_{j-1}, Z_{j-1}\big) \xrightarrow{x_j} \big(S_j, q_j, r_j, Y_j, T_j, Z_j\big)\Big)_{j\in\{1,\dots,m\}}$$

with $S_j, T_j \subseteq Q, q_j, r_j \in Q, Y_j \subseteq \overline{X}, Z_j \subseteq X$. Denote also $\tilde{q}_j = (S_j, q_j, r_j, Y_j, T_j, Z_j)$. Consider further $a = (a_1, \dots, a_{2n}) \in \mathbb{N}^{2n}$ a tuple which is accepted by ρ.

Let us identify in this run the moments where the 0-transitions take place – denote them $p_1, \dots, p_k$, with $k \geq 0$ (there may be no 0-transition at all). Hence,

$$\big(S_{j-1}, q_{j-1}, r_{j-1}, Y_{j-1}, T_{j-1}, Z_{j-1}\big) \xrightarrow{0} \big(S_j, q_j, r_j, Y_j, T_j, Z_j\big), \text{ for } j \in \{p_1, \dots, p_k\}$$
$$\big(S_{j-1}, q_{j-1}, r_{j-1}, Y_{j-1}, T_{j-1}, Z_{j-1}\big) \xrightarrow{1}$$
$$\big(S_j, q_j, r_j, Y_j, T_j, Z_j\big), \text{ for } j \in \{1, \dots, m\} \setminus \{p_1, \dots, p_k\}$$

Hence, by construction, for $j \in \{p_1, \dots, p_k\}$, $q_j = r_{j-1}$, $S_j = S_{j-1} \cup \{q_{j-1}\}$, $T_{j-1} = T_j \cup \{r_j\}$, and $Y_{j-1} = \overline{X}$. Note also that $k \geq 1$.

Other important indices are those where the run passes through an accepting component – for each $i \in \{1, \dots, 2n\}$, denote j_i the moment when the run passes through Q_i. Observe that for each $i \in \overline{X}$, $j_i < p_1$ and $j_{n+i} \geq p_k$. Hence, for each $l \in \{1, \dots, k+1\}$, the l-th segment of the run that lies in between the $(p_{l-1}+1)$-th transition and the p_l-1-th transition contains only 1-transitions. Here we assume $p_0 = 0$ and $p_{k+1} = m$.

Consider the sequence composed of left active components in the l-th segment and of right active components in the $l+1$-th segment

$$\Big(\big(r_{j-1} \xrightarrow{x_j} r_j\big)_{j\in\{p_{l-1}+1,\dots,p_l-1\}}, r_{p_l} \xrightarrow{0} q_{p_l+1} = r_{p_l}, \big(q_{j-1} \xrightarrow{x_j} q_j\big)_{j\in\{p_l+1,\dots,p_{l+1}-1\}}\Big)$$

If we were in the strictly non-elastic case, this sequence would have given a run in $\mathcal{A}$ accepting some tuple b_l, such that $b^1 \odot \dots b^{k+1} = a$. This is not exactly what happens in our case: this sequence is a *segment* of an accepting run for such tuples b_l, as it only identifies the part of b^l in which its components b^l_i with $i \in \overline{X}$ lie. The rest of the accepting run must be recovered from the history and the prophecy components, and this is what we will do in the sequel.

Let us give first a consistent notation for our run segment:

$$q^l_j = \begin{cases} r_j & \text{for } j \in \{p_{l-1}+1, \dots, p_l - 1\}, \\ q_j & \text{for } j \in \{p_l, \dots, p_{l+1} - 1\} \end{cases}$$

We will extend this segment for all $j \in \{p_{l+1}, \dots, m\}$ by induction on j, by choosing, in the prophecy component S_j, a continuation of the segment built until $j-1$ – that is, a state q^l_j with $q^l_{j-1} \to q^l_j$. This choice is always possible by construction, and there are three possible cases:

1. $j = p_{l+1}$ (which is the initial case for the induction). In this case, by construction, $S_j = S_{j-1} \cup \{q_{j-1}\}$. But $q_{j-1} = q^l_{j-1}$; therefore, we put $q^l_j = q_{j-1} = q^l_{j-1}$.
2. $j = p_{l'}$ for some $l' \geq l+2$, hence $x_j = 0$. As above, we have $S_j = S_{j-1} \cup \{q_{j-1}\}$. By induction, we have that $q^l_{j-1} \in S_{j-1}$, hence we may put $q^l_j = q^l_{j-1} \in S_j$.
3. $x_j = 1$. In this case, by construction, for all $q \in S_{j-1}$ there exists $q' \in S_j$ with $q \xrightarrow{1} q'$. Since $q^l_{j-1} \in S_{j-1}$, this property assures us that we may choose a $q^l_j \in S_j$ with $q^l_{j-1} \xrightarrow{1} q^l_j$.

Similarly, we extend our segment for all $j \in \{0, \ldots, p_{l-1}\}$ by "inverse" induction on j, by choosing, this time, $q^l_{j-1} \in T_{j-1}$ such that $q^l_{j-1} \to q^l_j$:

1. $j = p_{l-1}$. We have that $T_{j-1} = T_j \cup \{q_j\}$ and $q_j = q^l_j$; therefore, we put $q^l_{j-1} = q_j = q^l_j$.
2. $j = p_{l'}$ for some $l' \leq l-2$, hence $x_j = 0$. As above, we have $T_{j-1} = T_j \cup \{q_j\}$. By induction, we have that $q^l_j \in T_j$, hence we may put $q^l_{j-1} = q^l_j \in T_{j-1}$.
3. $x_j = 1$. In this case, by construction, for all $q \in T_i$ there exists $q' \in T_{j-1}$ with $q \xrightarrow{1} q'$. Since $q^l_j \in T_j$, this property assures us that we may choose a $q^l_{j-1} \in T_{j-1}$ with $q^l_{j-1} \xrightarrow{1} q^l_j$.

We hence get a run $\rho^l = \big(q^l_j\big)_{j \in \{1,\ldots,m\}}$ in $\mathcal{A}$ in which some states are repeated. We will show that this run is accepting.

Observe first that, by construction, for each $i \in X$ and $j \in \{1, \ldots, m\}$ with $i \in Z_j \setminus Z_{j-1}$ we have that $S_j \cup T_j \cup \{q_j, r_j\} \in Q_i \cap Q_{n+i}$. On the other hand, since ρ is accepting, we have $Z_0 = \emptyset$ and $Z_m = X$, and $Z_{j-1} \subseteq Z_j$ for all $j \in \{1, \ldots, m\}$. Hence, for each $i \in X$ there must exist some $j \in \{1, \ldots, m\}$ with $i \in Z_j \setminus Z_{j-1}$. But then, for each $l \in \{1, \ldots, k\}$, $q^l_j \in S_j \cup T_j \cup \{q_j, r_j\}$, hence $q^l_j \in Q_i = Q_{n+i}$, which means that ρ^l passes through all accepting components with indices in X.

Now, for each $l \in \{1, \ldots, k\}$ and each $i \in Y_{p_{l-1}}$ we have that $q^l_{p_{l-1}} = q_{p_{l-1}} \in Q_{n+i}$ and $q^{l+1}_{p_{l-1}} = r_{p_{l-1}} \in Q_i$. On the other hand, for each $i \in \overline{X}$ and $j \in \{1, \ldots, m\}$ with $i \in Y_j \setminus Y_{j-1}$ we have that $q_j \in Q_{n+i}$ and $r_j \in Q_i$. But $Y_{p_l-1} = \overline{X}$, hence, for each $i \in \overline{X} \setminus Y_{p_{l-1}}$ there must exist an $j \in \{p_{l-1}, \ldots, p_l - 1\}$ such that $i \in Y_j \setminus Y_{j-1}$. This implies that $q^l_j = q_j \in Q_{n+i}$ and $q^{l+1}_j \in Q_i$ for each $i \in \overline{X}$.

In other words, the above argument shows that all ρ^l with $l \in \{2, \ldots, k\}$ are accepting, and that ρ^1 passes through all accepting components Q_i with $i \in X \cup (\overline{X}+n) \cup (X+n)$, and that ρ^{k+1} passes through all accepting components Q_i with $i \in X \cup \overline{X} \cup (X+n)$. It only remains to identify the points of passage of ρ^1 through accepting sets of indices in $\overline{X}$, and the points of passage of ρ^{k+1} through accepting sets of indices in $(\overline{X}+n)$. But these are exactly the points j_i, with $i \in \overline{X} \cup (\overline{X}+n)$, where ρ itself passes through $\tilde{Q}_i$: by construction, for each $i \in \overline{X}$, we have that $\tilde{q}_{j_i} \in \tilde{Q}_i$ and $\tilde{q}_{j_{n+i}} \in \tilde{Q}_{n+i}$. But this means that $q^1_{j_i} = q_{j_i} \in Q_{j_i}$ and $q^{k+1}_{j_{n+i}} \in Q_{j_{n+i}}$.

Finally, we need to prove that each ρ^l accepts a tuple b^l such that $b^1 \odot \ldots \odot b^{l+1} = a$. The construction of these tuples follows from the above proof that ρ^l is accepting. Their components are $b^l_i = b^j_{n+i} = j - \sum_{u \leq j} x_u$,

- For $l \in \{1, \ldots, k+1\}$ and $i \in X$ we must have $i \in Z_j \setminus Z_{j-1}$.
- For $l \in \{2, \ldots, k\}$ and $i \in Y_{p_{l-1}}$ we must have $j = p_{l-1}$.
- For $l \in \{2, \ldots, k\}$ and $i \in \overline{X} \setminus Y_{p_{l-1}}$ we must have $i \in Y_j \setminus Y_{j-1}$.
- For $l = 1$ and $i \in \overline{X}$ we must have $j = j_i$.
- For $l = k+1$ and $i \in \overline{X}+n$ we must have $j = j_i$.

This ends the proof of the reverse inclusion.

For the direct inclusion, take $a \in L_p(\mathcal{A})^{\circledast \geq 2}$, hence $a = b^1 \odot \ldots \odot b^{k+1}$ $(k \geq 1)$ and, for each $l \in \{1, \ldots, k+1\}$, take ρ^l an accepting run in $\mathcal{A}$ which is accepting b^l. We will choose these runs such that they have equal length, eventually by appending some loops in q_{**} at the end of the shorter ones. Hence, $\rho^l = \big(q^l_{j-1} \xrightarrow{1} q^l_j\big)_{j \in \{1,\ldots,m\}}$.

Remark 17 *Observe that the non-elasticity property for $L_p(\mathcal{A})^{\circledast}$, combined with the assumption that $L_p(\mathcal{A}) \subseteq \mathrm{Diag}_X \cap \overline{\mathrm{Diag}_{\overline{X}}}$, implies that, for each $l \in \{2, \ldots, k+1\}$, and each $i, i' \in \overline{X}$, and for all $l' \leq l$, $b^{l'-1}_{n+i} = b^{l'}_{i'} \leq b^l_i = b^{l-1}_{n+i}$. That is, the first moment ρ^l reaches an accepting component Q_i with $i \in \overline{X}$ happens after all runs $\rho^{l'}$ with $l' < l$ have passed through all accepting components with indices in $\overline{X}$.*

The idea is that each run ρ^l will be "left active" between the first and the last moment it encounters a component Q_i and "right active" between the first and the last moment it encounters a component Q_{n+i} with $i \in \overline{X}$. Hence, when the run ρ^l is left active, all runs $\rho^{l'}$ with $l' < l$ are in the history component, the run ρ^{l+1} is right active while all runs $\rho^{l'}$ with $l' > l+1$ are in the prophecy component. It then follows that we must insert 0-transitions when we have to shift ρ^l from left to right active. Accordingly, the run ρ that we will build for accepting a will be of length $m + k - 1$.

The run ρ starts in the state $\big(\emptyset, q_*, q_*, \emptyset, \{q_*\}, \emptyset\big)$ and its states are of the form $(S_j, q^l_j, q^{l+1}_j, Y_j, T_j, Z_j)$ with $S_j = \big\{q^{l'}_j \mid l' < l\big\}$ and $T_j = \big\{q^{l'}_j \mid l' > l+1\big\}$. Then for each $l \in \{2, \ldots, k-1\}$ and $j \in \{\min_{i\in\overline{X}} b^l_i + l, \ldots, \min_{i\in\overline{X}} b^{l+1}_i + l - 1\}$, the j-th transition in ρ is $(S_{j-1}, q^l_{j-1}, q^{l+1}_{j-1}, Y_{j-1}, T_{j-1}, Z_{j-1}) \xrightarrow{1} (S_j, q^l_j, q^{l+1}_j, Y_j, T_j, Z_j)$ where

$$S_j = \big\{q^{l'}_j \mid l' < l\big\} \qquad Y_j = Y_{j-1} \cup \big\{i \in \overline{X} \mid 1 \leq i \leq n, j = b^l_i - l + 1\big\}$$
$$T_j = \big\{q^{l'}_j \mid l' > l+1\big\} \qquad Z_j = Z_{j-1} \cup \big\{i \in X \mid j = b^l_i - l + 1\big\}$$

The "shifting" transitions are as follows: for each $l \in \{2, \ldots, k-1\}$, if we denote $j = \min_{i\in\overline{X}} b^l_i + l - 1$, then

$$(S_{j-1}, q^{l-1}_{j-1}, q^l_{j-1}, \overline{X}, T_{j-1}, Z_{j-1}) \xrightarrow{0} (S_j, q^l_j, q^{l+1}_j, Y_j, T_j, Z_j), \text{ where}$$

$$S_{j-1} = \big\{q^{l'}_j \mid l' < l-1\big\} \quad S_j = \big\{q^{l'}_j \mid l' < l\big\} \quad Y_j = \big\{i \in \overline{X} \mid 1 \leq i \leq n, j = b^l_i - l + 1\big\}$$
$$T_{j-1} = \big\{q^{l'}_j \mid l' > l\big\} \quad T_j = \big\{q^{l'}_j \mid l' > l+1\big\} \quad Z_j = Z_{j-1}$$

The segment $1 \leq j \leq \min_{i\in\overline{X}} b^2_i$ of ρ has transitions

$$(\emptyset, q^1_{j-1}, q^2_{j-1}, Y_{j-1}, T_{j-1}, Z_{j-1}) \xrightarrow{1} (\emptyset, q^1_j, q^2_j, Y_j, T_j, Z_j)$$

with $T_j = \big\{q^{l'}_j \mid l' > 2\big\}$, $Y_j = Y_{j-1} \cup \big\{i \in \overline{X} \mid 1 \leq i \leq n, j = b^1_i\big\}$ and $Z_j = Z_{j-1} \cup \big\{i \in X \mid j = b^1_i\big\}$.

Similarly, the segment $\min_{i\in\overline{X}} b^k_i \leq j \leq m$ of ρ has transitions

$$(S_{j-1}, q^1_{j-1}, q^2_{j-1}, Y_{j-1}, \emptyset, Z_{j-1}) \xrightarrow{1} (S_j, q^1_j, q^2_j, Y_j, \emptyset, Z_j)$$

with $S_j = \big\{q^{l'}_j \mid l' < k\big\}$, $Y_j = Y_{j-1} \cup \big\{i \in \overline{X} \mid 1 \leq i \leq n, j = b^k_i\big\}$ and $Z_j = Z_{j-1} \cup \big\{i \in X \mid j = b^k_i\big\}$.

Let us first show that ρ is indeed a run in $\mathcal{C}$ The only thing to be proved for this is that the 0-transitions are correctly linked with 1-transitions. More formally, we have to show that for each $l \in \{2, \ldots, k-1\}$, if $j = \min_{i\in\overline{X}} b^l_i + l - 2$, then $X_j = \overline{X}$. But this is a straightforward corollary of the observation 17 above.

To show that ρ is an accepting run, and that a is accepted by ρ we need to observe the passages of ρ through accepting states, namely that:

1. For each $i \in \overline{X}$, $(\emptyset, q^1_{b^1_i}, q^2_{b^1_i}, Y_{b^1_i}, T_{b^1_i}, Z_{b^1_i}) \in \tilde{Q}_i$, that is, $Y_{b^1_i} = \emptyset$ and $q^1_{b^1_i} \in Q_i$.
   But this follows easily from the remark 17, since any passage of ρ^1 through an index $i' \in \overline{X}$ precedes any passage of ρ^2 through an index $i' \in \overline{X}$.
2. For each $i \in \overline{X}$, $(S_{b^{k+1}_i+l-1}, q^k_{b^{k+1}_i+l-1}, q^{k+1}_{b^{k+1}_i+l-1}, Y_{b^{k+1}_i+l-1}, \emptyset, Z_{b^{k+1}_i+l-1}) \in \tilde{Q}_{n+i}$, that is, $Y_{b^{k+1}_i+l-1} = \overline{X}$ and $q^{k+1}_{b^{k+1}_i+l-1} \in Q_{n+i}$.
   This is again a corollary of the remark 17, since any passage of ρ^{k+1} through an index $n+i'$ with $i' \in \overline{X}$ is after any passage of ρ^k through an index $n+i''$ with $i'' \in \overline{X}$.
3. For each $i \in X$, if $j = b^1_i + l$ for some $1 \le l \le k$ then $(S_j, q^l_j, q^{l+1}_j, Y_j, T_j, Z_j) \in \tilde{Q}_i \cap \tilde{Q}_{n+i}$, that is, $i \in Z_j \setminus Z_{j-1}$. This is clear by construction of ρ.

These observations also imply that a is accepted by ρ, fact which ends the proof of the direct inclusion. Hence, for $L_p(\mathcal{A}) \subseteq \mathrm{Diag}_X \cap \overline{\mathrm{Diag}_{\overline{X}}}$ the theorem is proved.

For the most general case, we rely on the above construction and on the following decomposition of the transitive closure of a sum:

$$(A \cup B)^{\oplus} = A^{\circledast} \odot (B^{\oplus} A^{\oplus})^{\circledast} \odot B^{\circledast} \tag{4}$$

which is a corollary of the standard decomposition of a star of sums, $(A \cup B)^{\circledast} = (A^{\circledast} \odot B)^{\circledast} \odot A^{\circledast}$ [12]. Here we have denoted $A^{\oplus} = A \odot A^{\circledast}$.

Take $X, Y \subseteq \{1, \ldots, n\}$ with $X \cap Y = \emptyset$. We will actually show that, for any $2n$-automaton $\mathcal{A}$ whose language is non-elastic and with $L_p(\mathcal{A}) \subseteq \overline{\mathrm{Diag}}_X \cap \mathrm{Diag}_Y$, we may build a $2n$-automaton for $L_p(\mathcal{A})^{\oplus}$. We will prove this fact by inverse induction on $k = card(X \cup Y)$. The case $k = n$ is exactly the above construction, since this implies $Y = \overline{X}$.

For the induction case, suppose that the construction is available for all $X, Y \subseteq \{1, \ldots, n\}$ with $X \cap Y = \emptyset$ and $card(X \cup Y) \geq k+1$, and for all $2n$-automata $\mathcal{A}$ with $L_p(\mathcal{A}) \subseteq \overline{\mathrm{Diag}}_X \cap \mathrm{Diag}_Y$. Take then $X, Y \subseteq \{1, \ldots, n\}$ with $card(X \cup Y) = k$, $X \cap Y = \emptyset$ and $\mathcal{A}$ a $2n$-automaton with $L_p(\mathcal{A}) \subseteq \overline{\mathrm{Diag}}_X \cap \mathrm{Diag}_Y$. Take also $i \in \{1, \ldots, n\} \setminus (X \cup Y)$, and denote $L_1 = L_p(\mathcal{A}) \cap \mathrm{Diag}_{\{i\}}$ and $L_2 = L_p(\mathcal{A}) \cap \overline{\mathrm{Diag}}_{\{i\}}$. Hence $L = L_1 \cup L_2$.

The intersection construction allows us to build $2n$-automata for both L_1 and L_2. Note that $L_1 \subseteq \overline{\mathrm{Diag}}_X \cap \mathrm{Diag}_{Y \cup \{i\}}$, hence, by the induction hypothesis, we may compute $2n$-automata $L_1^{\oplus}$ and for $L_1^{\circledast}$. Observe that $L_1^{\oplus} \subseteq \overline{\mathrm{Diag}}_X \cap \mathrm{Diag}_{Y \cup \{i\}}$. Similarly, we have $L_2 \subseteq \overline{\mathrm{Diag}}_{X \cup \{i\}} \cap \mathrm{Diag}_Y$, hence we may compute $2n$-automata for $L_2^{\oplus}$ and for $L_2^{\circledast}$. We may also note that $L_2^{\oplus} \subseteq \overline{\mathrm{Diag}}_{X \cup \{i\}} \cap \mathrm{Diag}_Y$.

We then observe that $L_3 = L_1^{\oplus} \odot L_2^{\oplus} \subseteq \overline{\mathrm{Diag}}_X \cap \mathrm{Diag}_{Y \cup \{i\}}$, which means that, by the induction hypothesis we may also build an automaton for $L_3^{\circledast}$. But then Identity 4 ensures that the automaton for $L^{\circledast}$ can be computed by applying the union and composition constructions to the automata obtained so far.

7. Representing region matrices with n-automata

Our approach to computing the reachability relation of a timed automaton is to build a $2n$-automaton for each relation R_τ, then to apply the union/composition/star constructions for $2n$-automata. To show that this approach works, we have to adapt $2n$-automata to represent also regular sets of arbitrary region matrices. And we also have to show that the sets of region matrices representing relations $R_{qq'}$ in timed automata satisfy the nonelasticity property. These are the topics of this section.

The adaptation of n-automata for nonpoint region matrices is based upon the following idea: we represent each nonpoint n-region matrix with one of the point region matrices neighboring it, in pair with some information about the "direction" in which the representative is situated w.r.t. the original region matrix. For example, consider the region matrix $R = \begin{pmatrix} \{0\} &]2,3[&]1,2[\\ ]-3,-2[& \{0\} &]-1,0[\\ ]-2,-1[&]0,1[& \{0\} \end{pmatrix}$. Our idea is to represent R with the point region matrix $W = \begin{pmatrix} \{0\} & \{3\} & \{2\} \\ \{-3\} & \{0\} & \{-1\} \\ \{-2\} & \{1\} & \{0\} \end{pmatrix}$ in pair with the following "matrix of relational symbols": $M = \begin{pmatrix} `=` & `<` & `<` \\ `>` & `=` & `>` \\ `>` & `<` & `=` \end{pmatrix}$. The connection is the following: M_{12} is '$<$' because $R_{12} < W_{12}$, M_{22} is '$=$' because $R_{22} = W_{22}$, etc.

Two other representations of R by point region matrices can be found. Note that not all 9 combinations of upper and lower bounds of the intervals in R give a point region matrix – we have at most n neighboring point region matrices for each n-dimensional region matrix. Moreover, not all matrices of relational symbols may carry correct information regarding the "direction of approximation". The actual matrices we use are the following:

Definition 18 *An n-**relation** is an $n \times n$ matrix M over the set $\Gamma = \{`<`, `=`, `>`\}$ satisfying the following property: there exists no sequence of indices $(i_1, i_2, i_3, i_4 = i_1)$ with $i_1, i_2, i_3 \in \{1, \dots, n\}$ for which, for all $j \in \{1,2,3\}$, $M_{i_j,i_{j+1}} \in \{`<`, `=`\}$ and for some $j \in \{1,2,3\}$, $M_{i_j,i_{j+1}} = `<`$, The set of n-relations is denoted Γ_n.*

*For $M \in \Gamma_n$ and $X \subseteq \{1, \dots, n\}$, the X-**projection** of M is the $card(X)$-relation resulting by deleting from M the rows and columns that are not in X. Formally, for $X = \{k_1, \dots, k_m\}$, where $k_i < k_{i+1}$ for all $i \in \{1, \dots, m-1\}$, and for all $i, j \in \{1, \dots, m\}$, $\big(M\big|_X\big)_{ij} \stackrel{def}{=} M_{k_i,k_j}$.*

*Given an m-relation $M_1 \in \Gamma_m$, an n-relation $M_2 \in \Gamma_n$ and a positive integer $p \le \min(m,n)$, the p-**juxtaposition** of M_1 with M_2 is the set of $(m+n-p)$-relations:*

$$M_1 \blacksquare_p M_2 \stackrel{def}{=} \big\{M \in \Gamma_{m+n-p} \mid M\big|_{\{1,\dots,m\}} = M_1, M\big|_{\{m-p+1,\dots,m+n-p\}} = M_2\big\}.$$

*The **composition** of $M_1, M_2 \in \Gamma_{2n}$ is*

$$M_1 \odot M_2 \stackrel{def}{=} \big(M_1 \blacksquare_n M_2\big)\big|_{\{1,\dots,n\}\cup\{2n+1,\dots,3n\}}.$$

Note that $M_1 \blacksquare_p M_2 = \emptyset$ iff $M_1\big|_{\{m-p+1,\dots,m\}} \neq M_2\big|_{\{1,\dots,p\}}$.

We may easily show that relation composition is associative. The unit for composition on *sets* of $2n$-relations is

$$\mathbf{1}^{rel}_{2n} \stackrel{def}{=} \big\{M \in \Gamma_{2n} \mid \forall i,j \in \{1,\dots,n\}, M_{ij} = M_{n+i,j} = M_{i,n+j} = M_{n+i,n+j}\big\}.$$

Further, we may define the *star* of a set of $2n$-relations $\mathcal{M} \subseteq \Gamma_{2n}$ as $\mathcal{M}^{\circledast} = \bigcup_{k\in\mathbb{N}} \mathcal{M}^{k\odot}$, where $\mathcal{M}^{0\odot} = \mathbf{1}^{rel}_{2n}$ and $\mathcal{M}^{(k+1)\odot} = \mathcal{M}^{k\odot} \odot \mathcal{M}$. Then the structure $\mathcal{P}\big(\Gamma_{2n}, \cup, \odot, \circledast, \emptyset, \mathbf{1}^{rel}_{2n}\big)$ is a Kleene algebra.

Our representation of region matrices is the following:

Definition 19 *A tuple* $(W, M) \in \mathsf{PReg}_n \times \Gamma_n$ *is called an* ***n-region representation***. *The n-region matrix* ***represented*** *by* (W, M) *is denoted* $[W, M]$ *and defined as follows: for all* $i, j \in \{1, \ldots, n\}$,

$$[W, M]_{ij} \stackrel{def}{=} \begin{cases} \{W_{ij}\} & \textit{iff } M_{ij} = `=` \\ ]W_{ij}-1, W_{ij}[& \textit{iff } M_{ij} = `<` \\ ]W_{ij}, W_{ij}+1[& \textit{iff } M_{ij} = `>`. \end{cases}$$

The above definition would be incorrect unless we prove:

Proposition 20 *For each* $(W, M) \in \mathsf{PReg}_n \times \Gamma_n$, $[W, M]$ *has a nonempty semantics.*

The mapping $[\cdot] : \mathsf{PReg}_n \times \Gamma_n \to \mathsf{Reg}_n$ defines an equivalence relation on $\mathsf{PReg}_n \times \Gamma_n$ (the *kernel* of $[\cdot]$) denoted in the sequel $\equiv_n$ and defined as follows: $(W, M) \equiv_n (W', M')$ iff $[W, M] = [W', M']$. A set of n-region representations $\mathcal{M}$ is called **saturated** by $\equiv_n$ iff for each $(W, M) \in \mathcal{M}$, if $[W, M] = [W', M']$ for some $(W', M') \in \mathsf{PReg}_n \times \Gamma_n$ then also $(W', M') \in \mathcal{M}$. For each $\mathcal{W} \subseteq \mathsf{PReg}_n \times \Gamma_n$, we denote

$$[\mathcal{W}] \stackrel{def}{=} \big\{[W, M] \mid (W, M) \in \mathcal{W}\big\}.$$

We may now define projection, juxtaposition, composition and star on n-region representations:

$$(W, M)\big|_X \stackrel{def}{=} (W\big|_X, M\big|_X)$$
$$(W, M) \blacksquare_p (W', M') \stackrel{def}{=} \big\{W \blacksquare_p W', M'') \mid M'' \in M \blacksquare_p M'\big\}$$
$$(W, M) \odot (W', M') \stackrel{def}{=} \big((W, M) \blacksquare_n (W', M')\big)\big|_{\{1,\ldots,n\}\cup\{2n+1,\ldots,3n\}}$$

If we put $\mathbf{1}_{2n}^{preg} = \mathbf{1}_{2n}^{reg} \cap \mathsf{PReg}_{2n}$, then $\mathbf{1}_{2n}^{preg} \times \mathbf{1}_{2n}^{rel}$ is the unit for composition on sets of $2n$-region representations. Further, we may build the star of a set of $2n$-region representations $\mathcal{W} \subseteq \mathsf{PReg}_{2n} \times \Gamma_{2n}$ as usual, $\mathcal{W}^{\circledast} = \bigcup_{k\geq 0} \mathcal{W}^{k\odot}$, where $\mathcal{W}^{0\odot} = \mathbf{1}_{2n}^{preg} \times \mathbf{1}_{2n}^{rel}$ and $\mathcal{W}^{(k+1)\odot} = \mathcal{W}^{k\odot} \odot \mathcal{W}$.

Proposition 21 *For each* $(W, M) \in \mathsf{PReg}_n \times \Gamma_n$, $[W\big|_X, M\big|_X] = [W, M]\big|_X$.

For each two saturated sets $\mathcal{W}_1, \mathcal{W}_2 \subseteq \mathsf{PReg}_{2n} \times \Gamma_{2n}$, $\mathcal{W}_1 \odot \mathcal{W}_2$ *is a saturated set and* $[\mathcal{W}_1 \odot \mathcal{W}_2] = [\mathcal{W}_1] \odot [\mathcal{W}_2]$.

For each saturated set $\mathcal{W} \subseteq \mathsf{PReg}_{2n} \times \Gamma_{2n}$, $\mathcal{W}^{\circledast}$ *is saturated and* $[\mathcal{W}^{\circledast}] = [\mathcal{W}]^{\circledast}$.

Definition 22 *An* ***n-region automaton*** *is a tuple* $\mathcal{A} = (Q, \delta, Q_*, Q_1, \ldots Q_n, \lambda)$ *in which* Q, δ, *and* $Q_1, \ldots, Q_n$ *bear the same meaning and properties as in n-automata, while* $Q_* \subseteq Q$ *is a* set of initial states *and* $\lambda : Q \to \Gamma_n$ *is an* n-relation labeling function. *Additionally, it is required that if* $(q, q') \in \delta$ *then* $\lambda(q) = \lambda(q')$, *and that, for all* $q_* \in Q_*$, *for all* $i \in \{1, \ldots, n\}$ *and all* $q \in Q_i$, $(q_*, q_*), (q_*, q) \in \delta$.

The notion of accepting run is defined as in n-automata, with the only difference that it may start in any state from Q_*. Observe that, by definition, all the states in a run must be labeled with the same n-relation. An accepting run $\rho = (q_j)_{j\in\{0,\ldots,k\}}$ **accepts** an n-region representation (W, M) iff $\lambda(q_j) = M$ for all $j \in \{0, \ldots, k\}$ and there exists some n-tuple $(a_1, \ldots, a_n) \in \mathbb{N} \cap \|W\|$ such that $q_{a_i} \in Q_i$ for all $i \in \{1, \ldots, n\}$. We associate three languages to each n-region automaton $\mathcal{A}$:

• The n**-region representation language accepted by** $\mathcal{A}$ consists of the n-region representations accepted by some accepting run. and is denoted $L_{rep}(\mathcal{A})$.

• The **region language** of $\mathcal{A}$ is $L_{rgn}(\mathcal{A}) \stackrel{def}{=} \{[W, M] \mid (W, M) \in L_{rep}(\mathcal{A})\}$.

• The n**-tuple language** of $\mathcal{A}$ is $L_{tup}(\mathcal{A}) \stackrel{def}{=} \bigcup\{\|R\| \mid R \in L_{rgn}(\mathcal{A})\}$. We say that an n-automaton is **saturated** if its n-region representation language is saturated.

Observe that, for each two n-region automata $\mathcal{A}$ and $\mathcal{B}$, $L_{rgn}(\mathcal{A}) = L_{rgn}(\mathcal{B})$ iff $L_{tup}(\mathcal{A}) = L_{tup}(\mathcal{B})$, but it might be possible that $L_{rgn}(\mathcal{A}) = L_{rgn}(\mathcal{B})$ and $L_{rep}(\mathcal{A}) \neq L_{rep}(\mathcal{B})$, due to the possibility to represent the same n-region matrix by different n-region representations. But when $\mathcal{A}$ and $\mathcal{B}$ are saturated, then we also have $L_{rgn}(\mathcal{A}) = L_{rgn}(\mathcal{B})$ iff $L_{rep}(\mathcal{A}) = L_{rep}(\mathcal{B})$.

Proposition 23 *1) If $\mathcal{A}$ and $\mathcal{B}$ are two saturated n-region automata, then one can build a saturated n-region automaton for $L_{tup}(\mathcal{A}) \cup L_{tup}(\mathcal{B})$ and $L_{tup}(\mathcal{A}) \cap L_{tup}(\mathcal{B})$.*

2) For each saturated n-region automaton $\mathcal{A}$, and $X \subseteq \{1, \ldots, n\}$, $L_{tup}(\mathcal{A})\big|_X$ can be accepted by a saturated $card(X)$-automaton $\mathcal{B}$.

3) Given two saturated $2n$-region automaton $\mathcal{A}$ and $\mathcal{B}$, there exists a $2n$-region automaton $\mathcal{D}$ with $L_{tup}(\mathcal{D}) = L_{tup}(\mathcal{A}) \odot L_{tup}(\mathcal{B})$.

Theorem 24 *Given a saturated $2n$-region automaton $\mathcal{A}$, suppose that for all $k \in \mathbb{N}$, $L_{tup}(\mathcal{A})^{k\odot}$ is contains only non-elastic tuples. Then $L_{tup}(\mathcal{A})^{\circledast}$ can be recognized by a saturated $2n$-region automaton.*

The only thing that remains to be shown is that this theorem works for the relations $R_{qq'}$ in timed automata. To this end, remind that we have associated a conjunctive formula f_τ for each transition τ in a timed automaton. Each formula f_τ is representable by a $2n$-DBM $D(\tau)$. Let us denote $\mathcal{L}$ the union of the semantics of all $2n$-DBMs $D(\tau)$, over all possible transitions τ that may occur in some timed automaton.

Proposition 25 *$\mathcal{L}$ contains only non-elastic tuples and is closed under composition.*

Therefore, for any subset of $\mathcal{L}$ which is representable by a $(2n+2)$-region automaton we may apply Theorem 24. Observe that the closure under composition of $\mathcal{L}$ implies that all relations $R_{qq'}$ are included in $\mathcal{L}$.

8. Conclusions

We have presented a relational approach for computing reachability relations in timed automata, approach which is based upon representing clock relations by n-automata. We have seen that, contrary to DBMs, n-automata may represent “disjunctive” information, that is, formulas over diagonal clock constraints that utilize disjunctions. A nice feature of this approach is its modularity, which may help in building “parallel” model-checking algorithms.

Observe that there is no connection between the $2n$-automaton obtained for each relation $R_{qq'}$ in a timed automaton and the *region construction* of [1]. Each composition and star “forgets” the intermediary steps within a run, hence we cannot recover all the “discrete states” in the region automaton.

Unfortunately n-automata are essentially nondeterministic, hence the problem of finding small automata representations of timing constraints is somewhat related to the problem of finding small nondeterministic finite automata. In particular, the union construction of several n-automata has to be accompanied by a technique which identifies some states, in order to reduce the state space.

One direction of future research is to compare the non-elasticity property with the "reversal-bounded" properties that assure decidability of Turing machines [6].

Acknowledgments

The author wishes to thank Eugene Asarin and Oded Maler for their support in doing this research.

References

[1] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[2] E. Asarin, P. Caspi, and O. Maler. A Kleene theorem for timed automata. In *Proceedings of LICS'97*, pages 160–171, 1997.

[3] R. Bellmann. *Dynamic Programming*. Princeton University Press, 1957.

[4] J. Bengtsson, B. Jonsson, J. Lilius, and Wang Yi. Partial order reductions for timed systems. In *Proceedings of CONCUR'98*, volume 1466 of *LNCS*, pages 485–500, 1998.

[5] H. Comon and Y. Jurski. Timed automata and the theory of real numbers. In *Proceedings of CONCUR'99*, volume 1664 of *LNCS*, pages 242–257, 1999.

[6] Zhe Dang. Binary reachability analysis of pushdown timed automata with dense clocks. In *Proc. of CAV 2001*, volume 2102 of *LNCS*, pages 506–517, 2001.

[7] C. Dima. *An algebraic theory of real-time formal languages*. PhD thesis, Université Joseph Fourier Grenoble, France, 2001.

[8] C. Dima. Computing reachability relations in timed automata. In *Proceedings of LICS'02*, pages 177–186, 2002.

[9] S. Gaubert and Max Plus. Methods and applications of $(\max, +)$ linear algebra. In *Porceedings of STACS'97*, volume 1200 of *LNCS*, pages 261–282, 1997.

[10] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley/Narosa Publishing House, 1992.

[11] Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine. Integration graphs: a class of decidable hybrid systems. In *Proceedings of "Workshop on Theory of Hybrid Systems"*, volume 736 of *LNCS*, pages 179–208, 1992.

[12] D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Information and Computation*, 110:366–390, 1994.

[13] K. G. Larsen, Paul Petterson, and Wang Yi. Uppaal: Status & developments. In *Proceedings of CAV'97*, LNCS, pages 456–459, 1997.

[14] I. Meiri, R. Dechter, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, pages 61–95, 1991.

[15] S. Mukhopadhyay and A. Podelski. Beyond region graphs: Symbolic forward analysis of timed automata. In *Proceedings of FST&TCS'99*, volume 1738 of *LNCS*, pages 232–244, 1999.

[16] A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, 1951.

[17] S. Yovine. Model-checking timed automata. In *Lectures on Embedded Systems*, volume 1494 of *LNCS*, pages 114–152, 1998.