



Analysis of timed automata with guards in dioids algebra

Julien Niguez, Saïd Amari, Jean-Marc Faure

► To cite this version:

Julien Niguez, Saïd Amari, Jean-Marc Faure. Analysis of timed automata with guards in dioids algebra. WODES 2016, 13th International Workshop on Discrete Event Systems, May 2016, Xi'an, China. hal-01326722

HAL Id: hal-01326722

<https://hal.science/hal-01326722>

Submitted on 5 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Analysis of timed automata with guards in dioids algebra

Julien Niguez, Saïd Amari and Jean-Marc Faure

LURPA, ENS Cachan, Univ. Paris-Sud

Supmecca, Université Paris-Saclay

94235 Cachan, France

{julien.niguez;said.amari;jean-marc.faure}@ens-cachan.fr

Abstract— In this paper, we propose a new linear representation to model the behavior of Timed Automata with Guards (TAGs) using the formalism of dioids algebra. This linear modeling is used to define the parallel composition and properties of determinism for TAGs. The contribution is illustrated with an example of a jobshop to analyze the performances of this system.

Keywords — *Timed automata with guards; discrete event systems; linear representation; dioids algebra;*

I. INTRODUCTION

We are interested in this work to analyze the behavior of a class of timed Discrete Event Systems (DES). The timing aspect of DES can be handled through several modeling tools and analysis. In this paper, we will consider two complementary formalisms: approaches based on dioids algebra (in particular $(\max,+)$ and $(\min,+)$ algebra), and models based on timed automata.

On the first hand, $(\max,+)$ algebra (introduced in [3]) is particularly effective for the study of quantitative measures of DES, such as asymptotic performances, earliest or latest behaviors. However, this formalism is not adapted to dealing with conflicts or choices that are very common in DES. It has been shown in [6] that automata with multiplicities in $(\max,+)$ algebra, also called $(\max,+)$ automata, can be used to handle this issue. This approach combines ideas of automata with results on $(\max,+)$ algebra, to manage both logical and timing aspects of DES. $(\max,+)$ automata are the subject of numerous publications during the last years on performance evaluation ([6], [7]) and control ([9], [10], [12]). More recently, an approach presented in [4] proposes a new representation of $(\max,+)$ automata that takes advantages of both $(\max,+)$ and $(\min,+)$ algebras to determine the completion dates for the worst and the best schedules. Nevertheless, $(\max,+)$ automata only consider single weighted transitions (transitions associating a single duration to an event), which are not suited for industrial application. This comes from the fact that in practice, an event does not occur at the exact same time, and a task does not have an exact duration. It is more realistic to use intervals in order to describe these aspects.

On the other hand, timed automata are particularly efficient at handling intervals as durations of tasks and bounds of occurrences. There were first presented in [1] as timed graphs that use comparison between clocks values and transitions guards to rule the system evolutions. Based on works about timed transitions systems ([8]), a first extension

of these automata were proposed in [2]. In order to prevent that the system evolve into deadlocks caused by time elapsing (no more transition is validated for the clocks values), timed automata completed with state invariants, called Timed Automata with Guards (TAGs), were introduced in [6]. Finally, timed automata used in UPPAAL software were presented in [11]. UPPAAL automata are particularly powerful thanks to the high number of possibilities they offer (urgencies, synchronizations). However, to the best of our knowledge, none of these formalisms had been used as a base for a linear representation. Hence, it is not possible to use results of $(\max,+)$ algebra on these automata.

In order to fill the gap between timed automata and $(\max,+)$ based approaches, [13] proposed an utilization of techniques based on semirings of interval and linear algebra to model a class of timed automata (Interval Weighted Automata) in a linear manner. However, the clocks used in these automata are reset at every transition. This restriction is not suitable for modeling systems in which durations are stated for sequences instead of events.

This paper proposes a linear representation of TAGs by means of dioids, taking advantage of the modeling capacity of TAGs and the analysis possibilities of $(\max,+)$ automata at the same time.

The paper is organized as follows. In the following section, we recall the formalism of TAGs. A linear representation of TAGs is introduced in section III, and an application of this representation is presented in section IV. Finally, conclusions and possible future works are drawn up.

II. TIMED AUTOMATA WITH GUARDS

This section reminds the formalism of Timed Automata with Guards presented in [5] and provides some illustrative examples.

A. Definition of a TAG

Definition 1 [5]: A Timed Automaton with Guards, denoted by G , is a 7-tuple

$$G = (Q, \Sigma, Q_0, Q_m, Tra, Inv, C)$$

where:

- Q is the set of states;
- $Q_0 \subset Q$ is the set of initial states;
- $Q_m \subset Q$ is the set of final (or marked) states;
- Σ is a finite set of events;

- C is the set of clocks, c_1, \dots, c_n , with $c_i(t) \in \mathbb{R}^+$, $t \in \mathbb{R}^+$;
- Tra is the set of *timed transitions* of the automaton with

$$Tra \subseteq Q \times \mathcal{C}(C) \times \Sigma \times 2^C \times Q$$
 where $\mathcal{C}(C)$ is the set of admissible constraints for the clocks in the set C ;
- Inv is the set of *state invariants*, $Inv : Q \rightarrow \mathcal{C}(C)$;

The set Tra of timed transitions is to be interpreted as follows. If

$$(q_{in}, guard, e, reset, q_{out}) \in Tra$$

then there is a transition from q_{in} to q_{out} with the complete label

$$(guard, ; e, reset)$$

where $guard \in \mathcal{C}(C)$, $e \in \Sigma$, and $reset \subseteq C$.

The set of admissible clock constraints $\mathcal{C}(C)$ is specified as follows:

- If $I \subseteq \mathbb{R}^+$, then all conditions of the form $c_i(t) \in I$ are in $\mathcal{C}(C)$.
- If g_1 and g_2 belong to $\mathcal{C}(C)$, then $g_1 \wedge g_2$ belongs to $\mathcal{C}(C)$

Example 1: Fig. 1 depicts an example of an alarm modeled with a TAG, called G_1 , with a clock c_1 and a state invariant in state 2. This automaton will issue an event *alarm* if the duration between 2 consecutive occurrences of *msg* is smaller than 1 time unit. State 1 is considered as final. For the sake of simplicity, when the guard is $[0; +\infty[$, i.e. when an event can occur at any time, the notation $-$ is used in place of $[0; +\infty[$.

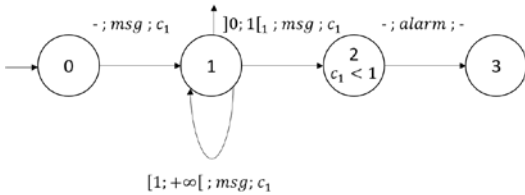
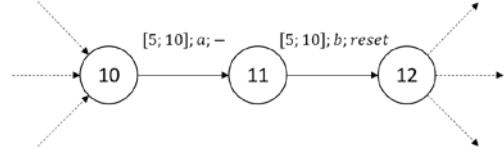


Figure 1: A Timed Automaton with Guards G_1

Remark 1:

- There is no need for the bounds of admissible clock constraints to be integer.
- All clocks are set to 0 when the system is initialized.
- Initial (final) states are indicated with incoming (outgoing) arrows. In Fig. 1, state 0 is initial and state 1 is final.
- The mechanism of *reset* allow modeling systems in which durations are stated for sequences of events. For example, the constraint “sequence *ab* must last at least 5 t.u. and at most 10 t.u.” can be modeled as follow. This kind of

constraint cannot be treated with automata used in [13].



Properties of TAGs, such as determinism and model execution are detailed in [5]

B. Parallel Composition

Definition 2 [5]: Consider two timed automata with guards

$$G_1 = (Q_1, \Sigma_1, Q_{0,1}, Q_{m,1}, Tra_1, Inv_1, C_1) \quad \text{and}$$

$$G_2 = (Q_2, \Sigma_2, Q_{0,2}, Q_{m,2}, Tra_2, Inv_2, C_2). \quad \text{The parallel composition of } G_1 \text{ and } G_2 \text{ is the automaton:}$$

$$G_{1||2} = \mathcal{Ac}(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, Q_{0,1} \times Q_{0,2}, Q_{m,1} \times Q_{m,2}, Tra_{1||2}, Inv_{1||2}, C_1 \cup C_2)$$

where \mathcal{Ac} corresponds to accessible transitions and states,

$$Inv_{1||2} : Q_1 \times Q_2 \rightarrow \mathcal{C}(C_{1||2}) = \mathcal{C}(C_1) \wedge \mathcal{C}(C_2)$$

with $Inv_{1||2}(q_1, q_2) = Inv_1(q_1) \wedge Inv_2(q_2)$, and $Tra_{1||2}$ is defined as follows:

$$Tra_{1||2} \subseteq (Q_1 \times Q_2) \times \mathcal{C}(C)_{1||2} \times (\Sigma_1 \cup \Sigma_2) \times 2^{C_1 \cup C_2} \times (Q_1 \times Q_2)$$

- For all $e \in \Sigma_1 \cap \Sigma_2$, if $(q_{i,in}, guard_i, e, reset_i, q_{i,out}) \in Tra_i$ for $i = 1, 2$, then

$$((q_{1,in}, q_{2,in}), guard_1 \wedge guard_2, e, reset_1 \cup reset_2, (q_{1,out}, q_{2,out})) \in Tra_{1||2};$$

- For all $e_1 \in \Sigma_1 \setminus \Sigma_2$ and $q_2 \in Q_2$, if $(q_{1,in}, guard_1, e_1, reset_1, q_{1,out}) \in Tra_1$, then

$$((q_{1,in}, q_2), guard_1, e_1, reset_1, (q_{1,out}, q_2)) \in Tra_{1||2};$$

- For all $e_2 \in \Sigma_2 \setminus \Sigma_1$ and $q_1 \in Q_1$, if $(q_1, q_{2,in}, guard_2, e_2, reset_2, q_{2,out}) \in Tra_2$, then

$$((q_1, q_{2,in}), guard_2, e_2, reset_2, (q_1, q_{2,out})) \in Tra_{1||2}.$$

Example 2: Recall the automaton G_1 of Fig.1. Suppose that a second TAG, called G_2 and shown in Fig. 2.1, was built in order to issue an event *alarm5* if two occurrences of events *msg* are more than 5 time units apart. The parallel composition of G_1 and G_2 , called $G_{1||2}$, is shown in Fig. 2.2. The set of transitions has been built exhaustively according to definition 2. Since clocks c_1 and c_2 reset when entering the state (1,B) as shown in the transition $(-; msg; c_1, c_2)$, conditions $c_1 \in]0; 1[$ and $c_2 \in]5; +\infty[$ of the transition $(]0; 1[\wedge]5; +\infty[; msg; c_1, c_2)$ cannot be satisfied. Hence, the state (2,D) and its successors are not reachable.

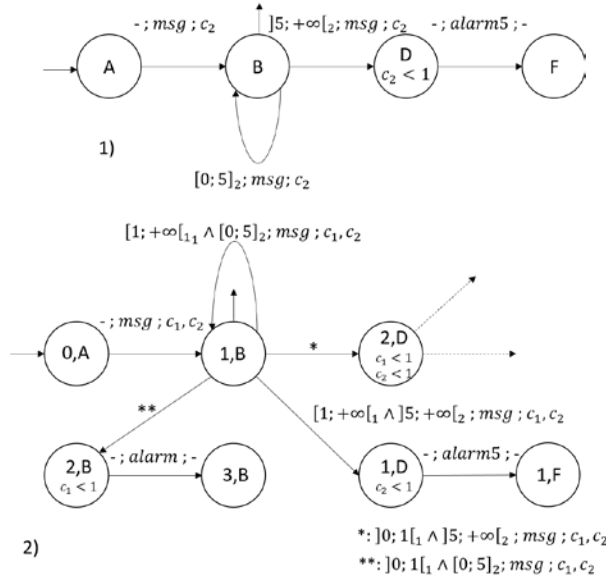


Figure 2 – 1. A Timed Automaton with Guards G_2 (top) - 2. Parallel composition $G_1 ||_2$ of G_1 and G_2 (bottom)

C. Case of single-clock systems

In most cases, for small-size systems, only one clock is sufficient. Concerning larger systems, they can be handled by using decentralized approaches, in which each sub-system is modeled with a single clock. In that specific case of single clock systems, the parallel composition could be simplified since the conjunction of two guards would become equivalent to the intersection of the intervals. If the result of that intersection is the empty set, then the guard can never be validated, and the associated transition can be deleted.

Example 3: Consider that the automata G_1 and G_2 from examples 1 and 2 share the same clock c . The automaton $G'_{1||2}$ shown in Fig. 3 is the result of the parallel composition.

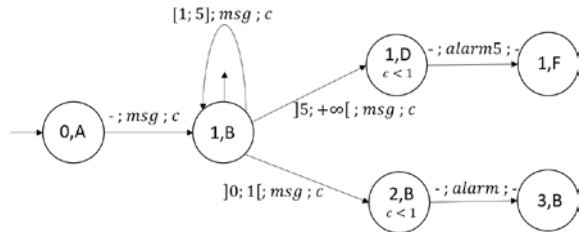


Figure 3 – TAG $G'_{1||2}$ for a single-clock case

III. LINEAR REPRESENTATION OF TIMED AUTOMATA WITH GUARDS USING DIOIDS ALGEBRA

This section presents our proposition for a linear representation of TAGs using dioids. In this section, we will assume that for all $q \in Q$, $Inv(q) = true$, that is, the linear notation proposed below does not consider the set of *state invariants* Inv . To maintain the coherence of the examples, state invariants from Fig. 1, 2 and 3 will be held by outgoing transitions of these states e.g. the state invariant $c < 1$ of the state (1,D) in Fig. 3 will become a guard in the transition from state (1,D) to state (1,F): $([0; 1]; alarm5; -)$. Moreover, we will only consider single-clock system in this section.

A. Definition

Consider a dioid $(\mathbb{D}, \oplus, \otimes, \epsilon, e)$, such that ϵ (resp. e) is the neutral element for \oplus (resp. \otimes).

The set $\mathbb{R} \cup \{-\infty\}$ with the maximum playing the role of addition and conventional addition playing the role of multiplication is a dioid, denoted \mathbb{R}_{Max} and usually called *(max,+) algebra*, with $e = 0$ and $\epsilon = -\infty$. Dioid \mathbb{R}_{Max} completed with $T = +\infty$ is denoted $\overline{\mathbb{R}}_{Max}$.

Similarly, the *(min,+) algebra* is the set $\mathbb{R} \cup \{+\infty\}$ with the minimum as addition, the conventional addition as multiplication, $e = 0$ and $\epsilon = +\infty$.

Definition 3: A TAG can be defined as a 7-tuple $(Q, \Sigma, Tra, \alpha, \beta, \mu_{Sup}, \mu_{Inf})$, called *linear representation*, where:

- Q is the set of states;
- Σ is a finite set of events;
- Tra is the set of *timed transitions* of the automaton;
- $\alpha \in \overline{\mathbb{R}}_{Max}^{1 \times |Q|}$, if $q \in Q_0$ then $\alpha_q = e$ else $\alpha_q = \epsilon$;
- $\beta \in \overline{\mathbb{R}}_{Max}^{|Q| \times 1}$, if $q \in Q_m$ then $\beta_q = e$ else $\beta_q = \epsilon$;
- $\mu_{Sup}: \Sigma \rightarrow \mathbb{D}^{|Q| \times |Q|}$,

$$[\mu_{Sup}(e_v)]_{q_{in}q_{out}} \triangleq \begin{cases} \sup(guard) & \text{if } (q_{in}, guard, e_v, reset, q_{out}) \in Tra \\ \epsilon & \text{else} \end{cases}$$

- $\mu_{Inf}: \Sigma \rightarrow \mathbb{D}^{|Q| \times |Q|}$,

$$\mu_{Inf}(e_v)_{q_{in}q_{out}} \triangleq \begin{cases} \inf(guard) & \text{if } (q_{in}, guard, e_v, reset, q_{out}) \in Tra \\ \epsilon & \text{else} \end{cases}$$

In other words, coefficients $[\mu_{Sup}(e_v)]_{ij}$ and $[\mu_{Inf}(e_v)]_{ij}$ correspond respectively to the upper and lower bounds of the intervals in *guard* for every $(q_{in}, guard, e_v, reset, q_{out}) \in Tra$.

Namely, for $\mathbb{D} = \overline{\mathbb{R}}_{Max}$, a TAG can be represented with two *(max,+)* automata as presented in [5]. These automata correspond to the extremal behavior of the system modeled: μ_{Sup} (resp. μ_{Inf}) describes the upper bound (resp. lower bound) of the system. For the sake of simplicity, we will use the notation $T = +\infty$ in the examples presented below.

Remark 2: it is possible to take into account initial and final delays through vectors α and β . $\alpha_{q_i} = n$ (resp. $\beta_{q_i} = n$) means that the state i is initial (final) with an initial (final) delay of n time unit. By convention, if a state i is not initial (final), $\alpha_{q_i} = \epsilon$ ($\beta_{q_i} = \epsilon$).

Example 4: Recall the automaton G_1 from Fig. 1. We had $Q = \{0, 1, 2, 3\}$, $\Sigma = \{msg, alarm\}$, $Q_0 = \{0\}$, $Q_m = \{1\}$. For this example, we obtain the following initial/final vectors and matrices (these four matrices represent the behavior at the latest and at the earliest of the TAG G_1):

$$\alpha = (e \in \epsilon) ; \beta = \begin{pmatrix} \epsilon \\ e \\ \epsilon \\ \epsilon \end{pmatrix} ;$$

$$\mu_{Sup}(msg) = \begin{bmatrix} \epsilon & \top & \epsilon & \epsilon \\ \epsilon & \top & 1 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix} ; \mu_{Sup}(alarm) = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 1 \\ \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix} ;$$

$$\mu_{Inf}(msg) = \begin{bmatrix} \epsilon & 0 & \epsilon & \epsilon \\ \epsilon & 1 & 0 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix} ; \mu_{Inf}(alarm) = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 0 \\ \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix} ;$$

Similarly to *generalized daters* introduced in [6] in order to describe the evolution and the dynamic of (max,+) automata, we propose a definition for *generalized bounds* for a TAG.

Definition 4: Consider two vectors $x_{Sup}(w) \in \mathbb{R}_{Max}^{1 \times |Q|}$ and $x_{Inf}(w) \in \mathbb{R}_{Max}^{1 \times |Q|}$, $w \in \Sigma^*$, called *generalized bounds*, and defined as follows:

$$\text{In } \mathbb{R}_{Max}, \quad \begin{cases} x_{Sup}(\epsilon) = \alpha \\ x_{Sup}(we_v) = x_{Sup}(w) \otimes \mu_{Sup}(e_v) \end{cases}$$

$$\text{In } \mathbb{R}_{Max}, \quad \begin{cases} x_{Inf}(\epsilon) = \alpha \\ x_{Inf}(we_v) = x_{Inf}(w) \otimes \mu_{Inf}(e_v) \end{cases}$$

where ϵ is the *empty word* and $e_v \in \Sigma$.

$[x_{Sup}(w)]_q$ can be interpreted as the maximal date at which the state q can be reached following a sequence w from an initial state. By convention, $[x_{Sup}(w)]_q = \epsilon$ if the state q cannot be reached following the sequence w from an initial state. An application of *generalized bounds* is exposed in section IV.

B. Determinism

The determinism of a TAG can be expressed with its linear representation. However, we can find two definitions of the determinism:

- *Time-determinism* [5]: an automaton is deterministic if for all events at all states, the guards of the associated transitions are mutually exclusive.
- *Event-determinism* [10]: an automaton is deterministic if there is a single i such that $\alpha_i \neq \epsilon$ and for all $e_v \in \Sigma$, there exists at most one j such that $\mu(a)_{ij} \neq \epsilon$.

It is clear that the second definition is stronger than the first one. For the example in Fig 3. From the state (1B), there is more than one outgoing transition labelled with the event *msg*, which means that the automaton is not *event-deterministic*. However, the guards of these transitions do not overlap, hence the automaton is *time-deterministic*.

Property 1: if an automaton is *event-deterministic*, then for a sequence w , there exists at most one k such that $[x_{Sup}(w)]_k \neq \epsilon$ and $[x_{Inf}(w)]_k \neq \epsilon$.

Proof: Consider a TAG with its linear representation $G = (Q, \Sigma, Tra, \alpha, \beta, \mu_{Sup}, \mu_{Inf})$. We suppose that G is *event-deterministic*.

- $x_{Sup}(\epsilon) = \alpha$ and $x_{Inf}(\epsilon) = \alpha$. Since G is *event-deterministic*, there is a single i such that $\alpha_i \neq \epsilon$. Hence the property is true for $w = \epsilon$
- Suppose that the property is true for a sequence w (the calculation will only be shown for x_{Sup} since it works in a similar way for x_{Inf}). For $e_v \in \Sigma$,

$$x_{Sup}(we_v) = x_{Sup}(w) \otimes \mu_{Sup}(e_v)$$

$$[x_{Sup}(we_v)]_k = \bigoplus_i [x_{Sup}(w)]_i \otimes [\mu_{Sup}(e_v)]_{ik}$$

Since the property is true for w , there is at most one j such that $[x_{Sup}(w)]_j \neq \epsilon$. Hence

$$[x_{Sup}(we_v)]_k = [x_{Sup}(w)]_j \otimes [\mu_{Sup}(e_v)]_{jk}$$

Moreover, G is *event-deterministic*. Then there is at most one l such that $[\mu_{Sup}(e_v)]_{jl} \neq \epsilon$, hence, such that $[x_{Sup}(we_v)]_l \neq \epsilon$. The property is true for a sequence we_v .

The property is true for ϵ , and the heredity has been proved for any w . Hence, the property is proved by induction. ■

C. Parallel composition

Definition 5: Consider two TAGs $G_1 = (Q_1, \Sigma_1, Tra_1, \alpha_1, \beta_1, \mu_{1,Sup}, \mu_{1,Inf})$ and $G_2 = (Q_2, \Sigma_2, Tra_2, \alpha_2, \beta_2, \mu_{2,Sup}, \mu_{2,Inf})$. The *parallel composition* of G_1 and G_2 is the automaton $G_{1||2} = Ac(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, Tra_{1||2}, \alpha_{1||2}, \beta_{1||2}, \mu_{1||2,Sup}, \mu_{1||2,Inf})$ with:

$$\alpha_{1||2} \in \mathbb{R}_{Max}^{1 \times |Q_1 \times Q_2|},$$

$$[\alpha_{1||2}]_{q_i \times q_j} = \begin{cases} [\alpha_1]_{q_i} \oplus [\alpha_2]_{q_j} & \text{if } [\alpha_1]_{q_i} \neq \epsilon \text{ and } [\alpha_2]_{q_j} \neq \epsilon \\ \epsilon & \text{else} \end{cases}$$

$$\beta_{1||2} \in \mathbb{R}_{Max}^{|Q_1 \times Q_2| \times 1},$$

$$[\beta_{1||2}]_{q_i \times q_j} = \begin{cases} [\beta_1]_{q_i} \oplus [\beta_2]_{q_j} & \text{if } [\beta_1]_{q_i} \neq \epsilon \text{ and } [\beta_2]_{q_j} \neq \epsilon \\ \epsilon & \text{else} \end{cases}$$

- $\mu_{1||2,Sup} = \Sigma \rightarrow \mathbb{D}^{|Q_1 \times Q_2| \times |Q_1 \times Q_2|}$
- $\mu_{1||2,Inf} = \Sigma \rightarrow \mathbb{D}^{|Q_1 \times Q_2| \times |Q_1 \times Q_2|}$

Coefficients matrices of $\mu_{1||2,Sup}$ and $\mu_{1||2,Inf}$ can be calculated as follows:

- For all $e_v \in \Sigma_1 \cap \Sigma_2$,

$$\begin{aligned}
& \text{In } \bar{\mathbb{R}}_{Min}, \\
& [\mu_{1||2,Sup}(e_v)]_{(q_{1,in} \times q_{2,in}), (q_{1,out} \times q_{2,out})} \\
& = \begin{cases} \bigoplus_{i=1,2} [\mu_{i,Sup}(e_v)]_{q_{i,in}, q_{i,out}} & \text{if } [\mu_{i,Sup}(e_v)]_{q_{i,in}, q_{i,out}} \neq \epsilon \\ \epsilon & \text{else} \end{cases}
\end{aligned}$$

$$\begin{aligned}
& \text{In } \bar{\mathbb{R}}_{Max}, \\
& [\mu_{1||2,Inf}(e_v)]_{(q_{1,in} \times q_{2,in}), (q_{1,out} \times q_{2,out})} \\
& = \begin{cases} \bigoplus_{i=1,2} [\mu_{i,Inf}(e_v)]_{q_{i,in}, q_{i,out}} & \text{if } [\mu_{i,Inf}(e_v)]_{q_{i,in}, q_{i,out}} \neq \epsilon \\ \epsilon & \text{else} \end{cases}
\end{aligned}$$

- For all $e_1 \in \Sigma_1 \setminus \Sigma_2$ and $q_2 \in Q_2$,

$$[\mu_{1||2,Sup}(e_1)]_{(q_{1,in} \times q_2), (q_{1,out} \times q_2)} = [\mu_{1,Sup}(e_1)]_{q_{1,in}, q_{1,out}}$$

$$[\mu_{1||2,Inf}(e_1)]_{(q_{1,in} \times q_2), (q_{1,out} \times q_2)} = [\mu_{1,Inf}(e_1)]_{q_{1,in}, q_{1,out}}$$

- For all $e_2 \in \Sigma_2 \setminus \Sigma_1$ and $q_1 \in Q_1$,

$$[\mu_{1||2,Sup}(e_2)]_{(q_1 \times q_{2,in}), (q_1 \times q_{2,out})} = [\mu_{i,Sup}(e_2)]_{q_{2,in}, q_{2,out}}$$

$$[\mu_{1||2,Inf}(e_2)]_{(q_1 \times q_{2,in}), (q_1 \times q_{2,out})} = [\mu_{i,Inf}(e_2)]_{q_{2,in}, q_{2,out}}$$

Remark 3: Following that definition, built matrices of $\mu_{1||2,Inf}$ and $\mu_{1||2,Sup}$ contain values for transitions that cannot be reached from the initial states. In particular, if $\exists k \in [1; |Q_1 \times Q_2|]$ such that for all $e_v \in \Sigma_{1||2}$, $\alpha_k [\mu_{1||2,Sup}(e_v)]_{\cdot, k} = \epsilon$ and $[\mu_{1||2,Inf}(e_v)]_{\cdot, k} = \epsilon$ then the corresponding states cannot be reached from the initial states. In other words, if the k^{th} column of $\mu_{1||2,Sup}(e_v)$ and $\mu_{1||2,Inf}(e_v)$ are empty for all $e_v \in \Sigma_{1||2}$, then there will be no transition entering the corresponding state.

Hence, the notations $\dot{\mu}_{1||2,Sup}$ and $\dot{\mu}_{1||2,Inf}$ can be introduced, corresponding to $\mu_{1||2,Inf}$ and $\mu_{1||2,Sup}$ in which every column of ϵ is deleted.

Example 5: According to the definition 5, the dimension of matrices of $\mu_{1||2,Inf}$ and $\mu_{1||2,Sup}$ for the automaton resulting of the parallel composition of G_1 and G_2 is too important to be detailed in this paper ($|Q_1 \times Q_2| = 16$). However, it is possible to determine matrices $\dot{\mu}_{1||2,Sup}$ and $\dot{\mu}_{1||2,Inf}$, that are detailed in the following order: $A = \dot{\mu}_{1||2,Inf}(msg)$, $B = \dot{\mu}_{1||2,Sup}(msg)$, $C = \dot{\mu}_{1||2,Inf}(alarm)$, $D = \dot{\mu}_{1||2,Sup}(alarm)$, $E = \dot{\mu}_{1||2,Inf}(alarm5)$, $F = \dot{\mu}_{1||2,Sup}(alarm5)$.

$$A = \begin{bmatrix} \epsilon & 0 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 1 & 0 & \epsilon & 5 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}; B = \begin{bmatrix} \epsilon & T & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 5 & 1 & \epsilon & T & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix};$$

$$C = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 0 & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}; D = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & T & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix};$$

$$E = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 0 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}; F = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & T \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix};$$

It can be noted that defined as such, $\dot{\mu}_{Sup}$ and $\dot{\mu}_{Inf}$ may contain cycles of states that cannot be reached from the initial states. However, the corresponding coefficients will not impact the representation aside from the size of matrices.

IV. APPLICATION TO A JOBSHOP

We consider here a jobshop processing with two jobs J_1 and J_2 to illustrate our contribution (this example come from [7] and [10]). This jobshop will be studied using TAGs and their linear representations. However, for the sake of simplicity, the system is presented with a Petri net model in Fig. 4.

Job J_1 (resp. J_2) consists of three tasks a, b and c (d, e and f). *Processing times* are exposed in Fig. 4. We will consider that a task does not have an exact duration, hence, we will associate an interval to every transition of the system, that corresponds to the *processing times* $\pm 10\%$ (e.g. a processing time of 2 time unites will be represented as the interval $[1.8; 2.2]$). Two resources \mathcal{R}_1 and \mathcal{R}_2 are shared by tasks, such as \mathcal{R}_1 can be used by a, b, d and e , and \mathcal{R}_2 by b, c, e and f .

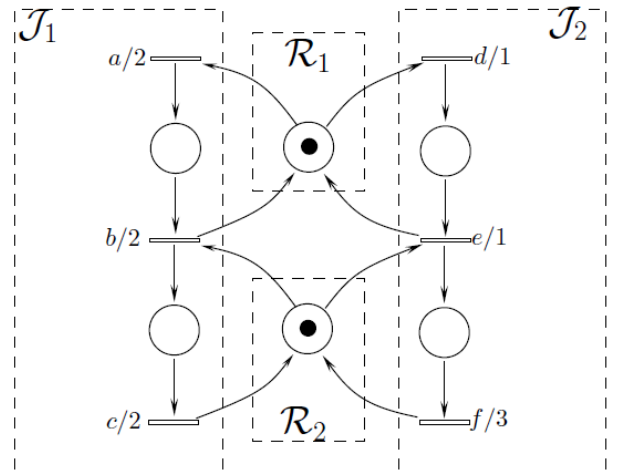


Figure 4 - Jobshop represented as a Petri net

The transformation of timed Petri nets into TAG is not addressed in this paper. For our application, the TAG model of Fig. 5 is inferred from the (max,+) automaton presented in

[illegible]

The linear representation of this TAG is:

$$\alpha = (\epsilon \epsilon \epsilon \epsilon \epsilon \epsilon \epsilon \epsilon \epsilon \epsilon), \quad \beta = (\epsilon \epsilon \epsilon \epsilon \epsilon \epsilon \epsilon \epsilon \epsilon \epsilon)^T,$$

$$\mu_{Sup}(a) \in \mathbb{D}^{|11| \times |11|}, \quad \mu_{Inf}(a) \in \mathbb{D}^{|11| \times |11|}, \quad \mu_{Sup}(b) \in \mathbb{D}^{|11| \times |11|},$$

$$\mu_{Inf}(b) \in \mathbb{D}^{|11| \times |11|}, \quad \mu_{Sup}(c) \in \mathbb{D}^{|11| \times |11|},$$

$$\mu_{Inf}(c) \in \mathbb{D}^{|11| \times |11|}, \quad \mu_{Sup}(d) \in \mathbb{D}^{|11| \times |11|}, \quad \mu_{Inf}(d) \in \mathbb{D}^{|11| \times |11|},$$

$$\mu_{Sup}(e) \in \mathbb{D}^{|11| \times |11|}, \quad \mu_{Inf}(e) \in \mathbb{D}^{|11| \times |11|},$$

$$\mu_{Sup}(f) \in \mathbb{D}^{|11| \times |11|}, \quad \mu_{Inf}(f) \in \mathbb{D}^{|11| \times |11|}.$$

Three of these matrices are detailed as an example. The other are not shown for the sake of clarity.

It is possible to determine *generalized bounds* (def. 4) for processing times of jobs J_1 and J_2 :

- $$\begin{aligned} x_{Sup}(abc) &= \alpha \otimes \mu_{Sup}(a) \otimes \mu_{Sup}(b) \otimes \mu_{Sup}(c) \\ x_{Sup}(abc) &= (\epsilon \in \epsilon \in \epsilon \in 6.6 \epsilon \in \epsilon \in \epsilon) \\ x_{Inf}(abc) &= \alpha \otimes \mu_{Inf}(a) \otimes \mu_{Inf}(b) \otimes \mu_{Inf}(c) \\ x_{Inf}(abc) &= (\epsilon \in \epsilon \in \epsilon \in 5.4 \epsilon \in \epsilon \in \epsilon) \end{aligned}$$

- Processing time of one job \mathcal{J}_2 :

$$\begin{aligned} x_{Sup}(def) &= \alpha \otimes \mu_{Sup}(d) \otimes \mu_{Sup}(e) \otimes \mu_{Sup}(f) \\ x_{Sup}(def) &= (\epsilon \in \epsilon \in \epsilon \in \epsilon \in \epsilon \in 5.5 \in \epsilon) \\ x_{Inf}(def) &= \alpha \otimes \mu_{Inf}(d) \otimes \mu_{Inf}(e) \otimes \mu_{Inf}(f) \\ x_{Inf}(def) &= (\epsilon \in \epsilon \in \epsilon \in \epsilon \in \epsilon \in 4.5 \in \epsilon) \end{aligned}$$

It can be noted that proposition 1 is respected for both sequences: since the TAG is *event-deterministic, generalized*

Representation of TAG in Fig. 5:

[illegible]

[illegible]

[illegible]

Another analysis can be done on the completion date for a given number of jobs, whatever the kind of job. Jobs \mathcal{J}_1 and \mathcal{J}_2 both consist of three tasks, which may be executed in parallel (e.g. tasks a and c do not use the same resource, hence, the $(n+1)^{\text{th}}$ job \mathcal{J}_1 can start before the end of the n^{th} job \mathcal{J}_1). For this reason, jobs may overlap and the duration of a succession of n job could be different from n times the duration of a single job.

For the completion date of 10 jobs:

- In the worst case, we found that the interval of the completion date is $[37.8; 46.2]$ time units that corresponds to a succession of 10 jobs \mathcal{J}_1 .

- In the best case, we found that the interval of the completion date is $[36.9; 45.1]$ time units that corresponds to an alternation of $\mathcal{J}_2\mathcal{J}_1$.

We managed here to get bounds that estimate the possible durations of sequences in the best and the worst cases, where [4] only provides a single value for the same durations. This is a clear improvement in the way that in a real application, the completion date of a succession of tasks cannot be exactly evaluated since the duration of a single task may vary depending to the environment of the system.

V. CONCLUSIONS

The paper has presented a new formal modeling of Timed Automata with Guards (TAGs) by dioids algebra. A linear representation of the behavior of TAGs is given. The definitions of a parallel composition and properties concerning the determinism of TAGs are proposed. This linear representation of TAGs is applied on a jobshop system to evaluate the temporal performances. In future works, an extension of the formalism to multiple-clocks systems could be considered. It would be also interesting to use this linear modeling of TAGs for diagnosis and fault-tolerant control of discrete event systems.

REFERENCES

- [1] R. Alur, C. Courcoubetis and D. Dill, "Model-checking for real-time systems", in Proceedings of 5th Annual IEEE Symposium on Logic in Computer Science, LICS'90, pages 414-425, 1990.
- [2] R. Alur and D. Dill, "A theory of timed automata", Theoretical computer science, 126(2):193-235, April 1994.
- [3] F. Baccelli, G. Cohen, G.-J. Olsder and J.-P. Quadrat, Synchronization and Linearity, Wiley, 1992.
- [4] R. Boukra, S. Lahaye and J.-L. Boimond, "New representations for $(\max,+)$ automata with applications to performance evaluation and control of discrete event systems", Discrete Event Dynamic Systems, 25(1-2):195-322, Springer US, 2013.
- [5] C.G. Cassandras and S. Lafortune, Introduction to Discrete event Systems, 2nd ed., Springer, 2008.
- [6] S. Gaubert, "Performance Evaluation of $(\max,+)$ Automata", IEEE Transactions on Automatic Control, 40(12):2014-2025, 1995
- [7] S. Gaubert and J. Mairesse, "Modeling and Analysis of Timed Petri nets using Heaps of Pieces", IEEE Transactions on Automatic Control, 44(4):683-698, 1999.
- [8] T. A. Henzinger, Z. Manna and A. Pnueli, Timed transition systems. In LNCS, Real Time : Theory in Practice, 600:226-251, 1992.
- [9] J. Komenda, S. Lahaye and J.-L. Boimond, "Supervisory control of $(\max,+)$ automata: a behavioral approach", Discrete Event Dynamic Systems, 91(4):525-549, Springer US, December 2009.
- [10] S. Lahaye, J. Komenda and J.-L. Boimond, "Compositions of $(\max,+)$ automata", 11th international workshop on discrete event systems, WODES 2012, Guadalajara, Mexico, October 3-5 2012.
- [11] P. Pettersson, Modeling and Verification of Real-Time Systems Using Timed Automata : Theory and practice, PhD thesis, Department of Computer Systems, Uppsala University, 1999.
- [12] R. Su, J. van Schuppen and J. Rooda, "The synthesis of time optimal supervisors by using heaps-of-pieces", IEEE Transactions on Automatic Control, 57(1):105-118, 2012.
- [13] J. Komenda, S. Lahaye and J.-L. Boimond, "Synchronous composition of interval weighted automata", 10th international workshop on discrete event systems, WODES 2010, Berlin, Germany, August 30- September 1 20