Control of (max,+) Automata: a single step approach

Jan Komenda, Sébastien Lahaye, and Jean-Louis Boimond

Abstract— **Control of (max,+) automata is studied within a behavioral framework. The classical tensor product of their linear representations and its generalized version extends the parallel composition of logical automata to (max,+) automata. In terms of behaviors (formal power series) these correspond to Hadamard product and a generalized version of it is studied in this paper. Supervisory control theory based on the generalized version of Hadamard product has an advantage that both logical and timing aspects can be captured at the same time using residuation theory of (multivariable) formal power series. Rationality as an equivalent condition to realizability of the resulting controller series is discussed.**

I. INTRODUCTION

(Max,+) automata have been introduced by S. Gaubert in [6] as (possibly nondeterministic) weighted automata with weights (multiplicities) in the $(\mathbb{R}\cup\{-\infty\}, \max, +)$ semiring.

An important class of Timed Discrete Event (dynamical) Systems (TDES), where both synchronization of tasks and resource sharing (choice) take place can be represented by (max,+) automata.

Recently, we have proposed two approaches to control (max,+) automata: an automaton or state-based one [11] and a behavioral (or formal power series) approach in [12].

The definition of parallel composition of weighted automata from [2] has been used in [12] for supervisory control of (max,+) automata. This composition corresponds to the tensor product in terms of linear representation in the $(\mathbb{R}\cup\{-\infty\}, \max, +)$ semiring. The controlled (closed-loop) system is given by the parallel composition of the controller automaton with the plant automaton.

In terms of behaviors, tensor product of (max,+) automata (strictly speaking of their linear representations) corresponds to Hadamard product of series. However, in the general case with uncontrollable events that can neither be forbidden and can nor be postponed (delayed), the approach proposed in [12] is not very elegant, because logical and timing aspects of control are separated: first supremal controllable sublanguage of the support of specification series is constructed and only in the second phase timing aspects are considered using residuation of Hadamard product, *i.e.* Hadamard inversion. This is not very elegant and suffers also from a computational viewpoint.

In this paper both logical and timing aspects of supervisory control are captured at the same time by generalizing parallel composition. This generalized version of the parallel composition of (max,+) automata, that we propose to take care of uncontrollable events, admits a similar representation (tensor product of the linear representations). Within a behavioral (formal power series) framework the parallel composition we propose corresponds to a generalized version (distinguishing uncontrollable events) of Hadamard product. Control with respect to the just in time criterion is then based on the residuation of generalized Hadamard product of formal power series.

This paper is a natural continuation and extension of results presented in [12]. Because we provide a single, closed expression, *i.e.* a formula for computation of controller series based on residuation of generalized Hadamard product of formal power series, the tedious two steps approach of [12] is then avoided.

This paper is organized as follows. In the next section basic algebraic preliminaries are recalled. In Section III we recall the definition of (max,+) (weighted) automata and propose a generalized parallel composition of (max,+) automata, which is applied to their supervisory control. Section IV is devoted to the main result of the paper: supervisory control of (max,+) automata is proposed, where both timing and logical aspects are handled at the same time. Rationality of the resulting controller series, *i.e.* of residuated series of the (generalized) Hadamard product is also discussed in Section IV. Conclusion is given in Section V.

II. ALGEBRAIC PRELIMINARIES

An *idempotent semiring* (also called dioid) is a set $\mathcal{D}$ endowed with two inner operations denoted $\oplus$ and $\otimes$. The addition $\oplus$ is commutative, associative, has a unit element ε, *i.e.* $\varepsilon \oplus a = a$ for each $a \in M$, and is idempotent: $a \oplus a = a$ for each $a \in M$. The multiplication $\otimes$ is associative, has a unit element e, and distributes over $\oplus$. Moreover, ε is absorbing for $\otimes$, *i.e.* $\forall a \in M: \ a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$.

In any dioid, a natural order is defined by: $a \preceq b \Leftrightarrow a \oplus b = b$. A dioid $\mathcal{D}$ is complete if each subset A of $\mathcal{D}$ admits a least upper bound denoted $\bigoplus_{x\in A} x$, and if $\otimes$ distributes with respect to infinite sums. In particular, $T = \bigoplus_{x\in\mathcal{D}} x$ is the greatest element of $\mathcal{D}$. In a complete dioid, the greatest lower bound, denoted by $\wedge$, always exists; $a \wedge b = \bigoplus_{x\preceq a, x\preceq b} x$.

The simplest examples of dioids are number dioids such as $\mathbb{R}_{\max} = (\mathbb{R}\cup\{-\infty\}, \max, +)$ with idempotent addition, denoted by $\oplus$: $a \oplus b = \max(a, b)$, and conventional addition playing the role of multiplication, denoted by $a \otimes b$ (or ab when unambiguous). If we add $T = +\infty$ to this set, the resulting dioid is complete and denoted by $\overline{\mathbb{R}}_{\max}$.
Matrix dioids are introduced in the same manner as in the conventional linear algebra. The (max,+) identity matrix of $\mathbb{R}_{\max}^{n\times n}$ is denoted by E.
Let us denote by $\mathbb{N}$ the set of natural numbers with zero. In complete dioids the star operation can be introduced by the

formula

$$a^* = \bigoplus_{n\in\mathbb{N}} a^n,$$

where by convention $a^0 = e$ and $a^n = a \otimes a^{n-1}$ for any a.

Theorem 2.1 (see [3]): Let $\mathcal{D}$ be a complete dioid, $x, a, b \in \mathcal{D}$. Equation

$$x = x \otimes a \oplus b, \tag{1}$$

admits $b \otimes a^*$ as the least solution.

We recall basic notions and results of residuation theory which allows defining 'pseudo-inverses' of some isotone maps (f is isotone if $a \preceq b \Rightarrow f(a) \preceq f(b)$) defined on ordered sets and, in particular, on dioids (see [3], §4.4.4).

Definition 2.1: An isotone map $f : \mathcal{D} \to \mathcal{C}$, where $\mathcal{D}$ and $\mathcal{C}$ are dioids, is said to be residuated if there exists an isotone mapping $h : \mathcal{C} \to \mathcal{D}$ such that

$$f \circ h \preceq Id_{\mathcal{C}} \text{ and } \ h \circ f \succeq Id_{\mathcal{D}}. \tag{2}$$

$Id_{\mathcal{C}}$ and $Id_{\mathcal{D}}$ are identity maps of $\mathcal{C}$ and $\mathcal{D}$ respectively. h is unique, it is denoted $f^\sharp$ and is called residual of f.
If f is residuated then $\forall y \in \mathcal{C}$, the least upper bound of subset $\{x \in \mathcal{D} | f(x) \preceq y\}$ exists and belongs to this subset. It is equal to $f^\sharp(y)$. Let us recall that multiplication in complete dioids is residuated. In particular, we have the following result.

Theorem 2.2: In a complete dioid $\mathcal{D}$ the isotone map $R_a : x \mapsto x \otimes a$ is residuated. The greatest solution of $x \otimes a \preceq b$ exists and is equal to $R_a{}^\sharp(b)$, also denoted $b \phi a$. This 'quotient' satisfies the following formulæ

$$(x \phi a) \otimes a \preceq x, \tag{f.1}$$
$$(x \otimes a) \phi a \succeq x. \tag{f.2}$$

Now we recall formal languages, formal power series and their properties. Formal languages over a finite alphabet A are subsets of the free monoid A^* of all finite sequences of words from A. The zero language is $0 = \{\}$, the unit language is $1 = \{\varepsilon\}$. We say that $u = u_1 \dots u_k \in A^*$ is a subword of $w \in A^*$ if there exists a factorization $w = w_1 u_1 w_2 \dots w_k u_k w_{k+1}$ with $w_i \in A^*$, $i = 1, \dots k+1$. The corresponding subword order on A^* is $u \preceq w$ iff u is a subword of $w \in A^*$.
In the sequel we will work with the dioid of formal power series in the noncommutative variables from A (transition labels) and coefficients from $\mathbb{R}_{\max}$ (corresponding to time). Formal power series form a dioid denoted $\mathbb{R}_{\max}(A)$, where addition and (Cauchy) multiplication are defined as follows. For two formal power series
$s = \oplus_{w\in A^*} s(w)w \in \mathbb{R}_{\max}(A)$ and $s' \in \mathbb{R}_{\max}(A)$,

$$s \oplus s' \triangleq \oplus_{w\in A^*}(s(w) \oplus s'(w))w \ ,$$

$$s \otimes s' \triangleq \oplus_{w\in A^*}(\oplus_{uv=w} s(u) \otimes s'(v))w.$$

This dioid is isomorphic to the dioid of generalized dater functions from A^* to $\mathbb{R}_{\max}$ *via* a natural isomorphism similarly as the dioid $\mathbb{Z}_{\max}(\gamma)$ of formal power series is isomorphic to the dioid of daters from $\mathbb{Z}$ to $\mathbb{Z}_{\max}$, used to study Timed Event Graphs (TEG) [3, §5.3]. This isomorphism associates to any $y : \ A^* \to \mathbb{R}_{\max}$ the formal power series $\oplus_{w\in A^*} y(w)w \in \mathbb{R}_{\max}(A)$. This dioid is complete if we work with series that admit coefficients in the completion of $\mathbb{R}_{\max}$, that is $\overline{\mathbb{R}}_{\max}$. We point out that for $s, s' \in \mathbb{R}_{\max}(A)$, $s \preceq s'$ with respect to the natural order on $\mathbb{R}_{\max}(A)$ means that $\forall w \in A^* : \ s(w) \preceq s'(w)$ in the sense of natural order on $\mathbb{R}_{\max}$, *i.e.* $s(w) \le s'(w)$ for all $w \in A^*$. The language $supp(s) = \{w \in A^* : \ s(w) \ne -\infty\}$ is called the support of the series s. It is known that a formal power series is recognizable by a finite weighted automaton iff it is rational, *i.e.* it can be formed by rational operations from polynomial series (those with finite support).

Besides Cauchy multiplication of series another multiplication (elementwise or word by word), called Hadamard product, will be needed and is defined by:

$$s, s' \in \mathbb{R}_{\max}(A), \ s \odot s' \triangleq \oplus_{w\in A^*}(s(w) \otimes s'(w))w.$$

The following proposition states that $H_y : \overline{\mathbb{R}}_{\max}(A) \to \overline{\mathbb{R}}_{\max}(A), s \mapsto s \odot y$ is residuated.

Proposition 2.3: The isotone mapping H_y: $\overline{\mathbb{R}}_{\max}(A) \to \overline{\mathbb{R}}_{\max}(A)$, $s \mapsto s \odot y$ is residuated and its residual is given by

$$H_y^\sharp(s)(w) = s(w) \phi y(w), \tag{3}$$

i.e. $H_y^\sharp(s) = \bigoplus_{w\in A^*}(s(w) \phi y(w))w$.

Let us mention that the claim of Proposition 2.3 can be strengthened, because Hadamard product admits an inverse, sometimes called Hadamard quotient of a formal power series. In this paper we use however a generalized version of Hadamard product, which is only residuated, and therefore we keep the notation of residuation theory.

Let us denote by P_c for a subset $A_c \subseteq A$ the natural projection from A^* to A_c^* that is morphism of monoids that from any string $w \in A^*$ projects away events from $A_u = A \setminus A_c$, *cf.* [17]. Formally, $P_c : A^* \to A_c^*$ it is defined as follows on events from A

$$P_c(a) = \begin{cases} a & \text{if } a \in A_c \\ \varepsilon & \text{if } a \in A \setminus A_c \end{cases}$$

and P_c is extented to words in such a way that P_c is concatenative: $P_c(a_1 \dots a_n) = P_c(a_1) \dots P_c(a_n)$. Similarly, P_c is extended to languages (subsets of A^*) in an obvious way: for $L \subseteq A^*$: $P_c(L) = \cup_{w\in L} P_c(w) \subseteq A_c^*$. In the sequel A_c and A_u play the role of controllable and uncontrollable events, respectively. Natural projections have many useful properties, among them we need the Lemma below.

Lemma 2.4: Let $A_c \subseteq A$ with the corresponding natural projection $P_c : \ A^* \to A_c^*$ and the inverse projection P_c^{-1} : $\mathrm{Pwr}(A_c^*) \to \mathrm{Pwr}(A^*)$. Then we have

(i) $P_c \circ P_c^{-1}$ is identity, *i.e.* $\forall L \subseteq A_c^* : \ P_c(P_c^{-1})(L) = L$
(ii) $\forall L \subseteq A^* : \ L \subseteq P_c^{-1}(P_c)(L)$

A notion of projection of formal power series will be needed.

Definition 2.2: For any formal power series $s = \oplus_{w\in A^*} s(w)w \in \mathbb{R}_{\max}(A)$ and $A_c \subseteq A$, with the associated natural projection $P_c: A^* \mapsto A_c^*$, we associate the projected series $P(s)$ given by the following coefficients:

$$P(s)(w) = s(P_c w).$$

Let us note the difference between $P(s)$ and the following formal power series: $\tilde{P}(s) = \oplus_{w \in A^*} s(w) P_c w \in \mathbb{R}_{\max}(A)$ that we have introduced in [11]. It is easily seen on the series supports (that are languages). While the operator $\tilde{P}(s)$ can only decrease the support, our operator $P(s)$ can only increase the support. In particular, let us notice that P has values in $\mathbb{R}_{\max}(A)$ and not in $\mathbb{R}_{\max}(A_c)$. For instance, if $A_c = \{a\} \subseteq \{a, u\} = A$ and $s = 1 \oplus 2a$ then $P(s) = 1u^* \oplus 2u^* a u^*$. Indeed, we have by definition $P(s)(\varepsilon) = P(s)(u) = P(s)(u^2) = \cdots = s(\varepsilon) = 1$ and similarly, $P(s)(w) = s(a) = 2$ for any $w \in u^* a u^*$. We have in general $P(s) \succeq s$ for any $s \in \mathbb{R}_{\max}(A)$. Hence, our operator $P: \mathbb{R}_{\max}(A) \to \mathbb{R}_{\max}(A)$ is not compatible with projection on languages (it is not the morphic extension of P_c).

Finally, we recall basic definitions of tensor products that will be used in section III.

If $A = (a_{ij})$ is a $m \times n$ matrix and B is a $p \times q$ matrix over a dioid, then their *Kronecker (tensor) product* $A \otimes^t B$ is the $mp \times nq$ block matrix

$$A \otimes^t B = \begin{bmatrix} a_{11} \otimes B & \cdots & a_{1n} \otimes B \\ \vdots & \ddots & \vdots \\ a_{m1} \otimes B & \cdots & a_{mn} \otimes B \end{bmatrix}$$

Otherwise stated, using the block form, the tensor product $C = A \otimes^t B$ of two square matrices $A = (a_{ij})_{i,j=1}^n$ and $B = (b_{kl})_{k,l=1}^m$ is the block $n.m \times n.m$ matrix, where the element indexed by ik and jl is given by $C_{ik,jl} = a_{ij} \otimes b_{kl}$.

III. Parallel composition of (max,+) automata

First we recall the definition of (max,+) automata, which are automata with multiplicities in the $\mathbb{R}_{\max}$ semiring [6].

Definition 3.1: A (max,+) automaton over an alphabet A is a quadruple $G = (Q, \alpha, t, \beta)$, where Q is a finite set of states, $\alpha: Q \to \mathbb{R}_{\max}$, $t: Q \times A \times Q \to \mathbb{R}_{\max}$, and $\beta: Q \to \mathbb{R}_{\max}$, called input, transition, and output delays, respectively.

The transition function associates to a state $q \in Q$, a discrete input $a \in A$ and a new state $q' \in Q$, an output value $t(q, a, q') \in \mathbb{R}$ corresponding to the $a-$transition from q to q' or $t(q, a, q') = \varepsilon$ if there is no transition from q to q' labelled by a. The real output value of a transition is interpreted as the minimal duration of the transition.

A (max,+) automaton is equivalently defined by a triple (α, μ, β), where $\alpha \in \mathbb{R}_{\max}^{1 \times Q}, \beta \in \mathbb{R}_{\max}^{Q \times 1}$ and μ is a morphism defined by:

$$\mu: A \to \mathbb{R}_{\max}^{Q \times Q},\ \mu(a)_{qq'} \triangleq t(q, a, q').$$

We will call such a triple a linear representation.

Note that the morphism matrix μ of a (max,+) automaton can also be considered as an element of $\mathbb{R}_{\max}(A)^{Q \times Q}$, *i.e.* $\mu = \oplus_{w \in A^*} \mu(w) w$ by extending the definition of μ from $a \in A$ to $w \in A^*$ using the morphism property

$$\mu(a_1 \dots a_n) = \mu(a_1) \dots \mu(a_n).$$

Since we want to extend the supervisory control techniques from logical to (max,+) automata, it is useful to formulate (max,+) automata in standard automata description (using initial and final states).

A (nondeterministic) (max,+) automaton over event alphabet A is the 4-tuple $G = (Q, q_0, Q_m, t)$, where Q is the set of states, q_0 is the initial state, Q_m is the subset of final or marked states, and $t: Q \times A \times Q \to \mathbb{R}_{\max}$ is the (possibly nondeterministic) transition function with inputs in A and outputs in $\mathbb{R}_{\max}$.

Note that the last definition does not consider initial delays, resp. final delays or strictly speaking these are only Boolean and equal to e *iff* the corresponding state is initial, resp. final.

The behaviour of the (max,+) automaton $G = (Q, \alpha, t, \beta)$ is given by the formal power series $l(G) \in \mathbb{R}_{\max}(A)$ defined for $w = a_1 \dots a_n \in A^*$ by

$$l(G)(w) = \max_{q_0, \dots, q_n \in Q} \alpha(q_0) \otimes \left[\sum_{i=1}^{n} t(q_{i-1}, a_i, q_i) \right] \otimes \beta(q_n). \tag{4}$$

Thus $l(G)(w)$ is the longest path along the word w starting at an initial state and ending at a final state, which corresponds to the completion time of the task w. Note that using the matrix formalism we have: $l(G)(w) = \alpha \otimes \mu(w) \otimes \beta$.

that

Any (max,+) automaton admits the following linear description in the dioid $\mathbb{R}_{\max}(A)$ of formal power series:

$$x = x\mu \oplus \alpha \tag{5}$$

$$y = x\beta, \tag{6}$$

where we also call $\mu = \bigoplus_{a \in A} \mu(a) a \in \mathbb{R}_{\max}(A)$ the morphism matrix.

Recall that according to theorem 2.1 the least solution to this equation is $y = \alpha \mu^* \beta$.

Let $A = A_c \cup A_u$ be the partition of A into disjoint subsets of controllable and uncontrolable events, respectively. The parallel composition below is defined as an extension of parallel composition (synchronous product) from logical to timed DES. The first automaton plays the role of the controller and the second is the system (to be controlled). We assume that the event sets of the controller and the plant automata are identical which is a standard assumption in supervisory control. In the case of controller defined only on a subalphabet it can be completed by inverse projection (*i.e.* by selflooping of all states with events not belonging to the subalphabet) to an automaton over the whole alphabet.

Definition 3.2: Consider the two following (max,+) automata corresponding to the controller and the system:

$$G_c = (Q_c, q_{c,0}, Q_m^c, t_c),\ G = (Q_g, q_{g,0}, Q_m^g, t_g). \tag{7}$$

Their *parallel composition*, modelling the system under control, is

$$\begin{aligned} G_c\|_{A_u}G &= (Q_c \times Q_g, q_0, Q_m, t) \\ \text{with } q_0 &= \langle q_{c,0}, q_{g,0}\rangle, \quad Q_m = Q_m^c \times Q_m^g, \end{aligned}$$

$$t(\langle q_c, q_g\rangle, a, \langle q_c', q_g'\rangle) =$$

$$\begin{cases} t_c(q_c, a, q_c') \otimes t_g(q_g, a, q_g'), & \text{if } a \in A_c \\ t_g(q_g, a, q_g'), & \text{if } a \in A_u \text{ and } q_c = q_c' \\ \varepsilon, & \text{if } a \in A_u \text{ and } q_c \neq q_c' \end{cases} \tag{8}$$

This definition can be viewed as an extension of prioritized synchronous composition from [8] from Boolean to the (max,+) case. Let us notice that this definition reflects the intuitive requirement that the controller automaton does not make any move if an uncontrollable event occurs: there are only selfloops of uncontrollable events. In fact we have two possibilities: either impose this restriction on the transition structure of the controller or define the parallel composition according to formula (8), where the cases $q_c = q_c'$ and $q_c \neq q_c'$ are distinguished. Actually, in the case $q_c = q_c'$ we have $t(\langle q_c, q_g\rangle, a, \langle q_c', q_g'\rangle) = t_g(q_g, a, q_g') = t_g(q_g, a, q_g') \otimes e = t_c(q_c, a, q_c') \otimes t_g(q_g, a, q_g')$ if we adopt the above described restriction on controller automata.

Controllable transitions (*i.e.* $t_g(q_g, a, q_g'),\ a \in A_c$) in the plant G can be in the composed system $G_c\|_{A_u}G$ both disabled (due to ε absorbing for multiplication : when the synchronizing transition of the controller is not defined $t_c(q_c, a, q_c') = \varepsilon$) and delayed (when $t_c(q_c, a, q_c') \geq 0$). The delay is added to the duration of the corresponding transition in $G_c\|_{A_u}G$. On the other hand, uncontrollable transitions (*i.e.* $t_g(q_g, a, q_g'),\ a \in A_u$) in the plant G can be in the composed system $G_c\|_{A_u}G$ neither disabled nor delayed.

Remark 3.1: There is the following interpretation of the parallel composition of a system with its controller. The controller is another (max,+)-automaton running in parallel (in a standard synchronous manner) with the system's automaton, that observes the generated events and either generates the same event as the controller, in which case it may delay the execution of the corresponding transition by the number of time units given by the weights of the transition in the controller (in case of a controllable event) or does not generate this event. In the latter case the event that was possible in the uncontrolled system is disabled in the parallel composition (this event should be controllable in accordance with definition). Uncontrollable events can neither be prevented from happening and can nor be delayed, the uncontrollable transition in the parallel composition inherits the duration from the original uncontrolled plant G.

Proposition 3.2: Consider two (max,+) automata and their linear representations:

$$G_c = (\alpha_c, \mu_c, \beta_c),\ G = (\alpha_g, \mu_g, \beta_g). \tag{9}$$

Their *parallel composition* in terms of linear representation is

$$\begin{aligned} G_c\|_{A_u}G &= (\alpha, t, \beta) \\ \alpha &= \alpha_c \otimes^t \alpha_g, \\ \forall a \in A_c:\ \mu(a) &= \mu_c(a) \otimes^t \mu_g(a), \\ \forall a \in A_u:\ \mu(a) &= E \otimes^t \mu_g(a), \\ \beta &= \beta_c \otimes^t \beta_g. \end{aligned}$$

Proposition 3.2 is useful for computing the behavior of the composed system consisting of a controller and a plant. Although we have formulated parallel composition in the state based framework (in order to make a clear connection with the classical supervisory control theory) the last proposition can be viewed as an equivalent definition of parallel composition for (max,+) automata in terms of their linear representations that admit nonzero initial and final delays from $\mathbb{R}_{\max}$.

IV. Application to supervisory control

Now parallel composition of Definition 3.2 is applied to the supervisory control of (max,+) automata.

We recall that the common event alphabets of the system and the controller is A. As usual in supervisory control, $A = A_c \cup A_u$ is partitioned into disjoint subsets of controllable events (which can be forbidden as well as delayed) and uncontrollable events (which can neither be forbidden nor delayed).

A behavioral framework is considered: instead of working with (max,+) automata we work with their behaviors: formal power series from $\overline{\mathbb{R}}_{\max}(A)$. This is quite natural, because control specifications of supervisory control are typically given by languages, here by formal power series. The resulting series corresponding to an optimal supervisor can then be realized by a (max,+) automaton, provided it is rational.

A two step procedure has been proposed in [12] that consists in separating the logical and timing aspects of control: first supremal controllable sublanguage of the specification support is computed and then timing aspects are considered (which amounts to $A_c = A$). In this paper we propose a more challenging approach and show how to handle the logical and timing aspects of the specification at the same time, within a single step procedure.

The relationship between tensor product and usual product of matrices, well known as the mixed product property will be useful:

Property 4.1: For matrices A, B, C, D of suitable dimensions over commutative dioid $\mathbb{R}_{\max}$ we have:

$$(A \otimes^t C) \otimes (B \otimes^t D) = (A \otimes B) \otimes^t (C \otimes D).$$

We need a formula for the behavior (*i.e.* formal power series) of parallel composition of the controller (max,+) automaton with the plant (max,+) automaton.

Theorem 4.2: The behavior of the parallel composition (according to Definition 3.2) is given by:

$$l(G_c\|G)(w) = l_c(P_c(w)) \otimes l_g(w).$$

By comparing the definition of Hadamard product with the formula of the last theorem we can view the right hand side as a kind of generalized Hadamard product (in presence of uncontrollable events). We propose the following definition.

Definition 4.1: Let $A = A_c \cup A_u$ with the associated natural projection $P_c : A^* \to A_c^*$. The generalized Hadamard product of two formal power series s and s', denoted $\odot_{A_u}$, is defined by $(s \odot_{A_u} s')(w) = s(P_c(w)) \otimes s'(w)$.

It follows from theorem 4.2 that

$$l(G_c \| G) = l(G_c) \odot_{A_u} l(G).$$

This can be applied to control of (max,+) automata in a behavioural framework.

Let y_{ref} be a specification series, the problem is to find the greatest controller series, denoted y_C such that $y_C \odot_{A_u} y \preceq y_{ref}$. Having in mind the meaning of order relation in $\overline{\mathbb{R}}_{\max}(A)$, one can give the following interpretations. Finding the greatest y_C, that is the greatest coefficients $y_C(w)$ for all w, and as a by-product the greatest coefficients $(y_C \odot_{A_u} y)(w)$, means that the controller will delay as much as possible the completion of the sequence of events w in the supervised system (whose behavior is given by $y_C \odot_{A_u} y$). In addition, since $y_C \odot_{A_u} y \preceq y_{ref}$, the completion date in the supervised system $(y_C \odot_{A_u} y)(w)$ is earlier than the completion date specified by $y_{ref}(w)$ for all sequence w. In other words, the considered control objective satisfies the just-in-time criterion, notably considered for the control of Timed Event Graphs (see for example [9], [10]).

Let us introduce the notation

$$H_y^{A_u} : s \mapsto s \odot_{A_u} y$$

for the right generalized Hadamard product.

Since $H_z^{A_u} : \overline{\mathbb{R}}_{\max}(A) \to \overline{\mathbb{R}}_{\max}(A)$ is again a residuated mapping (with its residuated mapping denoted by $(H_y^{A_u})^\sharp$), there exists the greatest y_C such that $H_y^{A_u}(y_c) \preceq y_{ref}$, namely $y_C^{opt} := (H_y^{A_u})^\sharp(y_{ref})$.

Proposition 2.3 has the following variant in presence of uncontrollable events ($A_u \neq \emptyset$).

Proposition 4.3: The mapping $H_y^{A_u} : \overline{\mathbb{R}}_{\max}(A) \to \overline{\mathbb{R}}_{\max}(A)$ is residuated and its residuted mapping is given by

$$(H_y^{A_u})^\sharp(s)(w) = \tag{10}$$

$$\begin{cases} \bigwedge_{u \in P_c^{-1}(w) \cap supp(y)}((s(u) \not{} y(u)), & \text{if } w \in A_c^* \\ T, & \text{if } w \notin A_c^* \end{cases}$$

We can check the correctness of this result by the following alternative approach. Using the following modified definition of projected formal power series $P_y : \overline{\mathbb{R}}_{\max}(A) \to \overline{\mathbb{R}}_{\max}(A)$ with

$$P_y(s)(w) = \begin{cases} s(P_c(w)), & \text{if } w \in supp(y) \\ \varepsilon, & \text{if } w \notin supp(y) \end{cases} \tag{11}$$

we have in fact $H_y^{A_u} = H_y \circ P_y$, *i.e.* $\forall s \in \overline{\mathbb{R}}_{\max}(A)$: $H_y^{A_u}(s) = H_y(P_y(s))$. This is because $\otimes$ is absorbing and hence for $w \notin supp(y)$ we can put $P_y(s)(w) = \varepsilon$ without modifying the Hadamard product $H_y^{A_u}(s)(w)$.

Proposition 4.4: Mapping P_y defined by (11) on complete dioids of formal power series is residuated with its residuated mapping given by

$$P_y^\sharp(s)(w) = \begin{cases} \bigwedge_{u \in P_c^{-1}(w) \cap supp(y)} s(u), & \text{if } w \in A_c^* \\ T, & \text{if } w \notin A_c^* \end{cases}$$

Using the well known formula from residuation theory $(H_y^{A_u})^\sharp = P_y^\sharp \circ H_y^\sharp$, it remains to substitute the formulae for $P_y^\sharp$ (prop. 4.4) and $H_y^\sharp$ (prop. 2.3). This yields to the same formula as the one obtained in proposition 4.3.

Remark 4.5: We point out the following analogy with the classical supervisory control theory. The residuated mapping $(H_y^{A_u})^\sharp(s)$ plays the role (*i.e.* is a generalization of) the supremal controllable sublanguage of specification (reference) series s with respect to the plant y and A_u. Indeed, $H_y^{A_u}(s)$ plays the role of infimal controllable superlanguage of the specification series s with respect to y and A_u. It is just an extension to the (max,+) case of the algebraic counterpart of supervised product defined by coinduction in [14]. Actually, if we denote in the classical supervisory control theory the operator $H_L(K) = \inf \mathrm{C}(K, L, A_u)$ the resulting closed-loop system, which corresponds to the infimal controllable superlanguage of the specification language K with respect to plant language L and A_u, then it can be shown that this mapping is residuated in the dioid of formal languages and its residuated mapping is nothing else but $H_L^\sharp(K) = \sup \mathrm{C}(K, L, A_u)$. The last proposition can then be viewed as a generalization of the formula for $\sup \mathrm{C}$ operator from Ramadge-Wonham theory.

Example 1: We consider a DES (*e.g.* a manufacturing system) in which three distinct tasks can be done. These tasks, labelled a, b and c, last respectively 3, 4 and 5 units of time. The system can perform the following sequences of tasks : a, ab, abc, $abcb$, $abcbc$, ... This system can be modeled by the (max,+) automaton G displayed on figure 1.(a). The behavior of G can be traduced by the following series in $\overline{\mathbb{R}}_{\max}(A)$:

$$y = 3a(9bc)^*(4b + e).$$

For instance, $y(ab) = 7$ means that the sequence ab will be completed at the earliest at date 7 (considering that the system starts to operate at time 0).

It is assumed that the start of tasks a and c can be delayed (we may decide to postpone the execution of these tasks when they should be performed) or even forbidden (their execution can be prevented). On the contrary, the task b can neither be delayed nor forbidden (this task starts as soon as it can be performed). Denoting $A = \{a, b, c\}$ the set of events (alphabet), we then have $A_c = \{a, c\}$ and $A_u = \{b\}$.

We would like that the system operates at the latest according to the following series:

$$y_{ref} = 4a \oplus 9ab \oplus 14abc.$$

This means that the sequences a, ab and abc should be completed at the latest at dates 4, 9 and 14 respectively. In addition, any other sequence of tasks should not happen. This series is recognized by the (max,+) automation G_{ref}

displayed in figure 1.(b).
In order to achieve this goal, we will apply the proposed supervisory control.

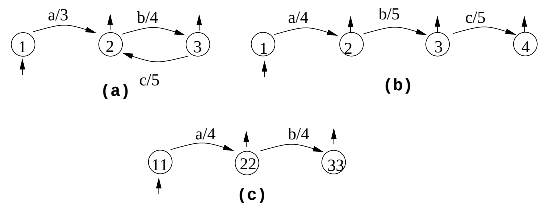

Fig. 1. G **(a)**, G_{ref} **(b)**, G_s **(c)**

Our approach dealing at the same time with both logical and timing aspects yields according to formula for $(H_y^{A_u})^\sharp$, $y_c^{opt}(a) = \min(y_{ref}(a) \not\phi y(a), y_{ref}(ab) \not\phi y(ab)) = \min(4 \not\phi 3, 8 \not\phi 7) = 1$, because $ab \in supp(y)$ and $P_c(ab) = a$, *i.e.* $\{u \in P_c P_c^{-1} a \cap supp(y)\} = \{a, ab\}$. For any other $w \in A^* \cap supp(y)$, we have $w \notin A_c^* \cap$ and $y_c^{opt}(w) = T$. It is then easy to check that the behavior of the system under control is $y_s = y_c^{opt} \odot y = \bigoplus_{w \in A^*} (y_c^{opt}(P_c(w)) \otimes y(w))\, w = 4a \oplus 8ab$. A (max,+) automation G_s which realizes y_s is displayed in figure 1.(c).

Let us remark finally that one should be aware of undecidability of equivalence (equality) of two rational formal power series with coefficients in $\mathbb{R}_{\max}$ [13]. Consequently, inequality is also undecidable. It is a serious problem as it is in general difficult or even impossible to verify that the synthesized controller satisfies the specification. On the other hand, our controller must satisfy the specification by construction, so there is no need for a systematic verification. Fortunately, there exist classes of formal power series, where equality is decidable, for (max,+) series equality (and inequality) is known to be decidable for series that are both (max,+) and (min,+) rational. Interestingly, it has been shown in [16] that these important classes of formal power series coincide with the so called *unambiguous series*. If we confine ourselves to this class of series there is no problem with decidability of inequalities.

Another important question is whether/when the resulting controller is rational. According to the above results this amounts to study the rationality of residuated mapping of Hadamard product. It turns out to be a difficult problem. The results of [16] are again helpful in this respect. Actually, the residuated mapping of Hadamard product can be viewed as the Hadamard product by the series that has inversed coefficients. More precisely, for any series $r \in \mathbb{R}_{\max}(A)$ let us denote by $C(r)$ the series with the coefficients $C(r)(w) = -r(w) \in \mathbb{R}_{\max}$. Then the residuated mapping of Hadamard product can be written by $H_y^\sharp(s)(w) = s(w) \not\phi y(w) = s \odot Cy$. Since Hadamard product is known to be a rational operation (realized by tensor product of linear representations, while realizable and rational formal power series coincide according to Schutzenberger's theorem), residuated mapping of Hadamard product is rational iff the "inversion" operator $C : \mathbb{R}_{\max}(A) \to \mathbb{R}_{\max}(A)$ preserves rationality. It has been shown in [16] that for a formal power series $s \in \mathbb{R}_{\max}(A)$ we have $C(s) \in \mathbb{R}_{\max}(A)$ iff s is unambigous. Moreover we recall that it is proven therein that $s \in \mathbb{R}_{\max}(A)$ is unambigous iff it is at the same time (max,+) and (min,+) rational.

V. Conclusion

We have presented a control mechanism for (max,+) automata based on the tensor product of their linear representation, *i.e.* Hadamard product of the corresponding formal power series. Both logical and timing aspects of their control have been studied using behavioral (formal power series) framework. In presence of uncontrollable events we have developped an approach based on a generalized version of Hadamard product and on direct application of residuation theory: both logical and timing aspects of supervisory control are handled at the same time.

In a future work we plan to develop decentralized control of (max,+) automata based on their synchronous product to be introduced.

References

[1] R. Alur and D. Dill. *The Theory of timed automata.* Theoretical computer science, 126:183-235, 1994.

[2] A. Arnold. *Finite Transition Systems. Semantics of Communicating Sytems*, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[3] Baccelli, F., G. Cohen, G.J. Olsder and J.-P. Quadrat (1992). *Synchronization and linearity-an algebra for discrete event systems.* New York, Wiley.

[4] J. Berstel and C. Reutenauer. *Rational series and their languages.* Berlin, Springer Verlag, 1988.

[5] S. Eilenberg. *Automata, Languages, and Machines*, Vol. A. Academic Press, New York, 1974.

[6] S. Gaubert. *Performance evaluation of (max,+) automata*, IEEE Trans. on Automatic Control, 40(12), pp. 2014-2025, 1995.

[7] S. Gaubert and J. Mairesse. *Modeling and analysis of timed Petri nets using heaps of pieces.* IEEE Trans. on Automatic Control, 44(4): 683-698, 1999.

[8] M. Heymann, *Concurrency and Discrete Event Control.* IEEE Control Systems Magazine, Vol. 10, No. 4, pp. 103-112, 1990.

[9] L. Houssin, S. Lahaye, J.-L. Boimond, *Just in Time Control of constrained (max,+)-Linear Systems.* Discrete Event Dynamic Systems, vol. 17, n°2, pp. 159-178, 2007.

[10] S. Lahaye, J.-L. Boimond, J.-L. Ferrier, *Just in Time Control of Time-Varying Discrete Event Dynamic Systems in (max,+) Algebra.* Int. J. of Production Research, vol. 46, no 19, pp. 5337-5348, 2008

[11] J. Komenda, M. Al Saba, and J.L. Boimond. Supervisory Control of (max,+) Automata: Timing Aspects. In Proceedings European Control Conference (ECC) 2007, Kos, Greece, July 2-5, 2007.

[12] J. Komenda, S. Lahaye, and J.L. Boimond. Control of (max,+) Automata: logical and timing aspects. In Proceedings of WODES 2008, Gothenburg, Sweeden, May 28-30, 2008.

[13] D. Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. Internat. J. Algebra Comput., 4, pp. 405 - 425, 1994.

[14] J. Komenda and J. H. van Schuppen: Control of Discrete-Event Systems with Partial Observations Using Coalgebra and Coinduction. Discrete Event Dynamical Systems: Theory and Applications 15(3), 257-315, 2005.

[15] S. Lombardy and J. Sakarovitch. *Derivatives of rational expressions with multiplicity.* Theoretical Computer Science 332, pp. 141-177, 2004.

[16] S. Lombardy and J. Mairesse. *Series which are both max-plus and min-plus rational are unambiguous*, RAIRO - Theoretical Informatics and Applications 40, pp. 1-14, 2006.

[17] F. Lin and W.M. Wonham, *I On Observability of Discrete-Event Systems*, Information Sciences, 44: 173-198, 1988.

[18] P.J. Ramadge and W.M. Wonham. The Control of Discrete-Event Systems. *Proc. IEEE*, 77:81-98, 1989.