Supervisory Control of (max,+) automata: a behavioral approach

Jan Komenda
Institute of Mathematics - Brno Branch, Czech Academy of Sciences
Zizkova 22, 616 62 Brno, Czech Republic
Email: komenda@ipm.cz
and
Sébastien Lahaye and Jean-Louis Boimond
LISA Angers,
62, Avenue Notre Dame du Lac, 49000 Angers, France,
Email: {lahaye, boimond}@istia.univ-angers.fr

A behavioral framework for control of (max,+) automata is proposed. It is based on behaviors (formal power series) and a generalized version of the Hadamard product, which is the behavior of a generalized tensor product of the plant and controller (max,+) automata in their linear representations.

In the tensor product and the Hadamard product, the uncontrollable events that can neither be disabled nor delayed are distinguished. Supervisory control of (max,+) automata is then studied using residuation theory applied to our generalization of the Hadamard product of formal power series. This yields a notion of controllability of formal power series as well as (max,+)-counterparts of supremal controllable languages. Rationality as an equivalent condition to realizability of the resulting controller series is discussed together with hints on future use of this approach.

1 Introduction

Supervisory control techniques proposed by Ramadge and Wonham in their seminal paper [23] have known rapid development. However, these have been proposed for purely logical discrete-event systems, where only ordering of discrete events, but not the actual timing of events and delays, is considered. Several attempts have been made to extend supervisory control theory to Timed Discrete Event (dynamical) Systems (TDES), starting from discrete time of [8], where time is represented by special discrete events. Later, supervisory control has been extended to general dense real-time systems like timed automata [25]. This approach is based on abstractions of timed automata into logical automata, called region automata.

An extension of the supervisory control approach to a class of TDES, modeled by (max,+) automata, is proposed in this paper. Our approach avoids any abstraction and is purely algebraic: it applies generalized inversion theory, called residuation theory, to semirings of formal power series.

(Max,+) automata model an important class of TDES, exhibiting both synchronization of tasks and resource sharing (choice) phenomena and moreover can be nondeterministic. They have been proposed by Gaubert in [10] as (possibly nondeterministic) weighted automata with weights (multiplicities) in the $(\mathbb{R} \cup \{-\infty\}, \max, +)$ semiring.

(Max,+) automata may also be viewed as a special class of timed automata, a general model for TDES introduced in [1]. On one hand, the modeling power of deterministic (max,+) automata in terms of timed automata is quite limited. Essentially, deterministic (max,+) automata can be viewed as one clock timed automata where the clock is reset after any transition. On the other hand, results of [11] show that (max,+) automata have a strong expressive power in terms of timed Petri nets: every 1-safe timed Petri net can be represented by a special (max,+) automaton, called heap model (or heap automaton). Hence heap models, *i.e.,* special non-deterministic (max,+) automata, are capable of modeling concurrent timed systems as timed Petri nets. This means that (max,+) automata can be used to describe various timed discrete-event systems such as those arising from manufacturing, transport or communication networks.

We have proposed a behavioral approach (based on formal power series) to supervisory control of (max+) automata in [15]. It consists of the parallel composition of controller and plant (max,+)-automata with uncontrollable events. Note that behavioural refers to the fact that formal power series are behaviors of weighted automata, which fact can be justified using theory of universal coalgebra. This is the same as in the classical Ramadge-Wonham approach where the supervisory control theory is based on formal languages that are behaviors of automata and not on state based models (*i.e.* automata themselves). It is however different from behavioural control setting of J.C. Willems, where behaviours are trajectories based on the timed and signal domains. Our systems are taken from computer science, so the philosophy is different. The controlled (closed-loop) system is given by the parallel composition of the controller automaton with the plant automaton.

The parallel composition used in this paper for supervisory control is the specialization to the $(\mathbb{R} \cup \{-\infty\}, \max, +)$ semiring of the one proposed for weighted automata in [2]. In terms of linear representations of (max,+) automata it corresponds to the tensor products in the $(\mathbb{R} \cup \{-\infty\}, \max, +)$ semiring, where the morphism matrix of the controller is replaced by the identity matrix for uncontrollable events which cannot be delayed nor disabled.

It is known that the Hadamard product of two series corresponds to the tensor product of automata [6] (strictly speaking of their linear representations). This means that the behavior of the tensor product of (max,+) automata is the Hadamard product of behaviors. More precisely, the proposed parallel composition takes the form of a generalized Hadamard product (distinguishing uncontrollable events). Control with respect to the just in time criterion is then based on the residuation of generalized Hadamard product of formal power series.

In the present paper, results of [15] are extended and presented in detail. Moreover, properties of this generalized Hadamard product (of a controller and plant formal power series) are investigated.

Controllability as an equivalent condition for attainability of a specification series as the prescribed behavior of the closed-loop system is studied using residuation theory of (multivariable) formal power series. A formula for computing (max,+)-counterparts of supremal controllable behaviors is proposed. A comparison with controllability of formal languages (from classical supervisory control theory) is given together with intuition behind timing aspects of controllability.

This paper is organized as follows. Necessary algebraic preliminaries are recalled in the next section. Parallel composition of (max,+) (weighted) automata is introduced in section 3. Then we recall the behavioral framework, where parallel composition of (max,+) automata corresponds to a generalized Hadamard product of formal power series. Section 4 is dedicated to the control problem and its optimal solution. In section 5 controllability and properties of controllable formal power series are investigated using residuation theory. In section 6, an example is studied to illustrate most of the results proposed in the paper. Conclusions with hints on future extensions of this work are finally given.

2 Preliminaries

In this section necessary algebraic concepts are recalled. The basic algebraic structure that is used across the paper is that of an idempotent semiring: *e.g.* formal languages, formal power series, transducers, number and matrix semirings.

2.1 Dioids and residuation

Definition 2.1. *An* idempotent semiring *(also called dioid) is a set $\mathcal{D}$ equipped with two binary operations: addition and multiplication. The addition $\oplus$ is commutative, associative, has a zero element ε (i.e., $\varepsilon \oplus a = a$ for all $a \in \mathcal{D}$), and is idempotent (i.e., $a \oplus a = a$ for all $a \in \mathcal{D}$). The multiplication $\otimes$ is associative, has a unit element e (i.e., $e \otimes a = a \otimes e = a$ for all $a \in \mathcal{D}$), and distributes over $\oplus$. Moreover, ε is absorbing for $\otimes$, i.e., $\forall a \in \mathcal{D}: \ a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$.*

In any dioid, a natural order $\preceq$ 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, $\top = \bigoplus_{x \in \mathcal{D}} x$ is the greatest element of $\mathcal{D}$. In a complete dioid, the greatest lower bound $\wedge$ always exists: $a \wedge b = \bigoplus_{x \preceq a, x \preceq b} x$.

Let us recall the dioid $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +)$ with maximum playing the role of addition $\oplus$: $a \oplus b = \max(a, b)$, and conventional addition playing the role of multiplication, denoted by $a \otimes b$ (or ab when unambiguous). Its complete version, completed with $\top = +\infty$, is denoted by $\overline{\mathbb{R}}_{\max}$. The reader is reminded that $\top \otimes \varepsilon = \varepsilon \otimes \top = \varepsilon$.

Operations with matrices are defined as in classical linear algebra. The (max,+) identity matrix of $\mathbb{R}_{\max}^{n \times n}$ is denoted by E.

Let $\mathbb{N}$ denote the set of natural numbers with zero. In complete dioids the star operation (sometimes referred to as *Kleene star*) can be introduced by the formula

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

where by convention $a^0 = e$ for any a.

Residuation theory (see [7]) allows defining 'pseudo-inverses' of isotone maps (f is *isotone* if $a \preceq b \Rightarrow f(a) \preceq f(b)$). An isotone map f is said to be *residuated* if $\forall y \in \mathcal{C}$, the least upper bound of subset $\{x \in \mathcal{D} | f(x) \preceq y\}$ exists and belongs to this subset. We recall from [3, §4.4] the following results.

Theorem 2.1. *An isotone map $f : \mathcal{D} \to \mathcal{C}$, where $\mathcal{D}$ and $\mathcal{C}$ are dioids, is residuated iff there exists an isotone mapping $h : \mathcal{C} \to \mathcal{D}$ such that*

$$f \circ h \preceq Id_{\mathcal{C}} \text{ and } \ h \circ f \succeq Id_{\mathcal{D}}. \tag{1}$$

$Id_{\mathcal{C}}$ and $Id_{\mathcal{D}}$ are identity maps of $\mathcal{C}$ and $\mathcal{D}$ respectively. The map h is unique, it is denoted $f^\sharp$ and is called residual of f.

Theorem 2.2. *A map $f : \mathcal{D} \to \mathcal{C}$ between two complete dioids is residuated iff*

(i) $f(\varepsilon) = \varepsilon$, and

(ii) f is lower semicontinuous, i.e., $f(\oplus_{x_i \in I} x_i) = \oplus_{x_i \in I} f(x_i)$ for all $I \subseteq \mathcal{D}$.

It is known that multiplication in complete dioids is residuated [3].

Theorem 2.3. *The isotone map $R_a : x \mapsto x \otimes a$ in a complete dioid $\mathcal{D}$ is residuated. The greatest solution of $x \otimes a \preceq b$ exists and is equal to ${R_a}^\sharp(b)$, also denoted $b \not{/} a$.*
This 'quotient' satisfies the following formulae

$$(x \not{/} a) \otimes a \preceq x, \qquad (f.1)$$
$$(x \otimes a) \not{/} a \succeq x. \qquad (f.2)$$

2.2 Dioid of formal power series $\overline{\mathbb{R}}_{\max}(A)$

Formal languages over an alphabet A are sets of finite sequences of letters (called words) from A. The set of all finite sequences of letters from A is denoted by A^*, which notation can be viewed as a special instance (applied to set A in the dioid of languages $(Pwr(A^*), \cup, .)$) of the star operation defined in subsection 2.1 on elements of complete dioids. Formal languages $L \subseteq A^*$ are endowed with union of languages playing the role of addition and concatenation playing the role of multiplication. The zero language is $0 = \{\}$, the unit language is $1 = \{\lambda\}$ with λ the empty (zero length) string. The dioid of formal languages is denoted by $(Pwr(A^*), \cup, .)$. A string $u = u_1 \dots u_k \in A^*$ is called a subword of $v \in A^*$ if there exists a factorization $v = v_1 u_1 v_2 \dots v_k u_k v_{k+1}$ with $v_i \in A^*,\ i = 1, \dots k+1$. The induced subword order on A^* is $u \preceq v$ iff u is a subword of $v \in A^*$.
The dioid of formal power series with variables from A and coefficients from $\mathbb{R}_{\max}$, endowed with pointwise addition and convolution multiplication, is denoted by $\mathbb{R}_{\max}(A)$. Thus, for $s = \oplus_{w \in A^*} s(w) w \in \mathbb{R}_{\max}(A)$ and $s' = \oplus_{w \in A^*} s'(w) w \in \mathbb{R}_{\max}(A)$, one has:

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

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

The dioid of formal power series is complete if we work with coefficients in $\overline{\mathbb{R}}_{\max}$. Note that for $s, s' \in \mathbb{R}_{\max}(A)$, $s \preceq s'$ (natural order on $\mathbb{R}_{\max}(A)$) amounts to $s(w) \leq s'(w)$ for all $w \in A^*$. The language $supp(s) = \{w \in A^* :\ s(w) \neq -\infty\}$ is called the support of the series s.

Another multiplication of formal power series of $\overline{\mathbb{R}}_{\max}(A)$ (element-wise or word by word), called Hadamard product, will be needed and is defined below.

Definition 2.2. *The Hadamard product of two series $s, s' \in \mathbb{R}_{\max}(A)$ is given by*

$$s \odot s' \triangleq \oplus_{w \in A^*} (s(w) \otimes s'(w)) w.$$

It has been shown in [15] that the Hadamard product is residuated.

Proposition 2.4. *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*

$$H_y^\sharp(s)(w) = s(w) \not{/} y(w), \qquad (2)$$

i.e., $H_y^\sharp(s) = \bigoplus_{w \in A^*} (s(w) \not{/} y(w)) w.$

Proof. From Theorem 2.1, we need to show that $H_y^\sharp$ defined by (2) is isotone and is such that inequalities (1) are satisfied. Since mapping $x \mapsto x \not{/} y$ on $\overline{\mathbb{R}}_{\max}$ is isotone, $H_y^\sharp$ is also isotone. Using successively $(f.1)$ and $(f.2)$ in dioid $\overline{\mathbb{R}}_{\max}$, we show the required inequalities:

$$(H_y \circ H_y^\sharp)(s) = \bigoplus_{w \in A^*} [(s(w) \not{/} y(w)) \otimes y(w)] w \preceq \bigoplus_{w \in A^*} s(w) w = s,$$

$$(H_y^\sharp \circ H_y)(s) = \bigoplus_{w \in A^*} [(s(w) \otimes y(w)) \not{/} y(w)] w \succeq \bigoplus_{w \in A^*} s(w) w = s.$$

□

Even more, the Hadamard product admits an inverse, which is known as the Hadamard quotient in the theory of formal power series over rings [24]. However, a generalized version of the Hadamard product, defined and used further in this paper, is only residuated. Hence, the notation of residuation theory is kept also for H_y.

Natural projections of languages are now recalled and extended to formal power series. The natural projection from A^* to A_c^*, where $A_c \subseteq A$ is denoted by P_c. It projects away from any word $w \in A^*$ the letters from $A_u = A \setminus A_c$, *cf.* [20].

Formally, $P_c : A^* \to A_c^*$ is defined as follows on letters 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 extended 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. The Lemma below recalls the known properties of natural projections [20].

Lemma 2.5. *Let $A_c \subseteq A$ with the corresponding natural projection $P_c : \ A^* \to A_c^*$ and the inverse projection $P_c^{-1} : \mathrm{Pwr}(A_c^*) \to \mathrm{Pwr}(A^*)$, one has*

(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. This notion is just an auxiliary concept that is useful in section 4 for better understanding the proposed solution to the supervisory control problem.

Definition 2.3. *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 $P_c : \ A^* \to A_c^*$, 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:
$\tilde{P}(s) = \oplus_{w \in A^*} s(w) P_c w$, *i.e.,* $\tilde{P}(s)(w) = \oplus_{u \in P_c^{-1}(w)} s(u)$. It is easily seen on the series supports (that are languages). For instance, if $A_c = \{a\} \subseteq \{a, u\} = A$ and $s = 1 \oplus 2a \oplus 2au$ then $\tilde{P}(s) = 1 \oplus 2a$, but $P(s) = 1u^* \oplus 2u^* a u^*$. Indeed, we have by definition $P(s)(\varepsilon) = s(\varepsilon) = 1$ as well as $P(s)(u^i) = s(\varepsilon) = 1$ for any $i \geq 1$, and similarly, $P(s)(w) = s(a) = 2$ for any $w \in u^* a u^*$. Hence, our operator $P : \ \mathbb{R}_{\max}(A) \to \mathbb{R}_{\max}(A)$ is not compatible with the projection on words and languages (it is not the morphic extension of P_c).

A modified projection for series will also be needed.

Definition 2.4. *Let $y \in \mathbb{R}_{\max}(A)$ and $A_c \subseteq A$ with the associated natural projection $P_c : \ A^* \to A_c^*$. We define the modified projection $P_y : \mathbb{R}_{\max}(A) \to \mathbb{R}_{\max}(A)$ by*

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

2.3 (Max,+) automata

Basic definitions and results on automata with multiplicities in the $\mathbb{R}_{\max}$ semiring, also called (max,+) automata, are recalled (see [10] for more details).

Definition 2.5. *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 initial, transition, and final delays, respectively.*

Let us note that instead of initial state we have an initial mapping that is called initial delay. Similarly, we have final delays that generalize the concept of final state subset. 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' labeled by a. The real output value of a transition is interpreted as the duration of this transition. An example of a (max,+) automaton is given in Section 6.

Since the state set Q plays only role of the dimension, a (max,+) automaton is determined 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:

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

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

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

Recall that μ has the important property of being finitely generated, because it is completely determined by its values on A. Thus, we have in fact $\mu^* = (\oplus_{a\in A}\mu(a)a)^*$. Since we are interested in behaviors of (max,+) automata that are given by $l = \alpha\mu^*\beta$ (see below), we abuse the notation and simply write $\mu = \oplus_{a\in A}\mu(a)a$.

We plan to extend the supervisory control techniques from logical to (max,+) automata. In that case, it is useful to formulate (max,+) automata in standard automata description (by replacing initial and final delays by initial and final states), that is as the 4-tuple $G = (Q, Q_0, t, Q_m)$, where Q is the set of states, $Q_0 \subseteq Q$ is the subset of initial states, $Q_m \subseteq Q$ 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 $\mathbb{R}_{\max}$.

Let us point out that this definition enables to only consider zero initial and final delays : the delay is equal to $e = 0$ if the state is initial or final, ε otherwise. This is because the subsets $Q_0 \subseteq Q$ and $Q_m \subseteq Q$ may be viewed as mappings $Q_0 \to \mathcal{B}$ and $Q_m \to \mathcal{B}$, where $\mathcal{B} = \{e, \varepsilon\}$ is the Boolean semiring.

Since (max,+) automata in state based form $G = (Q, Q_0, t, Q_m)$ are only used to better understand our generalization of the supervisory control approach [1] and our main results are formulated in terms of the standard linear description (*cf.* Definition 2.5), the generality is not lost.

Recall that a formal power series is recognized by a finite (max,+) automaton *iff* it is rational, *i.e.,* it can be formed by rational operations from polynomial series (those with finite support).

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

$$l(G)(w) = \max_{q_0,\dots,q_n\in Q} \alpha(q_0) \otimes \left[\sum_{i=1}^{n} t(q_{i-1}, a_i, q_i)\right] \otimes \beta(q_n). \tag{3}$$

[1] in particular the definition of parallel composition that is formulated in terms of more general linear description in Prop. 3.1

In words, $l(G)(w)$ is the maximal weights of paths labeled by w going from the initial state to a final state.

Remark 2.6. *The series $l(G) : A^* \to \mathbb{R}_{\max}$ is a dater [10]. We shall interpret $l(G)(w)$ as the time of completion of the sequence of events w, with the convention that $l(G)(w) = -\infty = \varepsilon$ if w does not occur. By specialization to "boolean" series with values in $\{\varepsilon, e\}$, we obtain the classical interpretation of Ramadge and Wonham theory, that is, $l(G)(w) \neq \varepsilon$ if w corresponds to an admissible behavior of the system.*
This way one can study logical aspects of $(\max, +)$ automata, while working with supports of formal power series corresponding to logical behaviors. We shall then consider series with boolean coefficients (in $\{\varepsilon, e\}$) instead of $\mathbb{R}_{\max}$ (any coefficient different from ε becomes e).

From its linear representation, the behavior of $(\max, +)$ automaton is given by

$$l(G)(w) = \alpha \otimes \mu(w) \otimes \beta,$$

that is,

$$l(G) = \alpha \otimes \mu^* \otimes \beta. \tag{4}$$

Similarly as timed event graphs are described by fixed point equations in the dioid of formal power series $Z_{\max}(\gamma)$, see [3, §5.3], any (max,+) automaton is described by the following fixed point equation in the dioid $\mathbb{R}_{\max}(A)$ of formal power series with non commutative variables from A:

$$\begin{cases} x &= x\mu \oplus \alpha, \\ y &= x\beta, \end{cases}$$

with $\mu = \bigoplus_{a \in A} \mu(a)a \in \mathbb{R}_{\max}(A)^{|Q| \times |Q|}$ the morphism matrix. It is known that the least solution to these equations is $y = l(G) = \alpha\mu^*\beta$.

3 Parallel composition of (max,+) automata

Parallel composition (or product) is defined below 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 a controller defined only on a subalphabet it can be completed by inverse projection (*i.e.,* by self-looping of all states with events not belonging to the subalphabet) to an automaton over the whole alphabet.

As usual in supervisory control, $A = A_c \cup A_u$ is the partition of the event set A into disjoint subsets of controllable and uncontrollable events, respectively. The parallel product is now defined.

Definition 3.1. *Consider the two following (max,+) automata corresponding to the controller and the system:*

$$G_c = (Q_c, Q_{c,0}, t_c, Q_m^c),\ G = (Q_g, Q_{g,0}, t_g, Q_m^g). \tag{5}$$

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

$$\begin{aligned} G_c \| G &= (Q_c \times Q_g, Q_0, t, Q_m) \\ \textit{with } Q_0 &= (Q_{c,0}, Q_{g,0}),\ \ Q_m = Q_c \times Q_m^g, \end{aligned}$$

$$t((q_c, q_g), a, (q_c', q_g')) =$$

$$\begin{cases} t_c(q_c, a, q_c') \otimes t_g(q_g, a, q_g'), & \text{if } a \in A_c, \\ t_g(q_g, a, q_g'), & \text{if } a \in A_u \text{ and } q_c = q_c', \\ \varepsilon, & \text{if } a \in A_u \text{ and } q_c \neq q_c'. \end{cases} \quad (6)$$

This definition can be seen as an extension of prioritized synchronous composition of [12] or [19] from Boolean to the (max,+) case. As in the classical supervisory control the controller cannot unmark the marked states of the original system: for any state that is marked in the original plant G and survives the logical supervision, the corresponding states in $G_c \| G$ are marked. This means that marked states of the controller do not play any role and may be ignored, which is expressed by $Q_m = Q_c \times Q_m^g$. In the sequel we can then assume that all states of the controller are marked without loss of generality.

Controllable transitions (*i.e.,* $t_g(q_g, a, q_g'),\ a \in A_c$) in the plant G can be disabled (when $t_c(q_c, a, q_c') = \varepsilon$) or delayed (when $t_c(q_c, a, q_c') > 0$) in the composed system $G_c \| G$.

On the other hand, uncontrollable transitions (*i.e.,* $t_g(q_g, a, q_g'),\ a \in A_u$) can neither be disabled nor delayed. It expresses the intuitive requirement that the controller automaton cannot disable an uncontrollable event that occurs in the plant.

The interpretation of the parallel composition of a system with its controller is as follows. The controller is another (max,+)-automaton running in parallel (in a standard synchronous manner) with the system's automaton. The controller (max,+)-automaton observes the events generated in the system and its state evolves consequently. From a given state q_c, if a transition associated to a controllable event a exists (*i.e.,* $\exists q_c'$ such that $t_c(q_c, a, q_c') \neq \varepsilon$), then this event is authorized in the system and the corresponding transition (in the system) is delayed by $t_c(q_c, a, q_c')$ units of time. In the latter case (*i.e.,* $\nexists q_c'$ such that $t_c(q_c, a, q_c') \neq \varepsilon$) the event that was possible in the uncontrolled system is disabled in the parallel composition. Uncontrollable events can neither be prevented from happening nor be delayed, the uncontrollable transition in the parallel composition inherits the duration from the original uncontrolled plant G.

3.1 Linear representation of a parallel composition

In the next proposition, the linear representation of the parallel composition of (max,+) automata is presented. The tensor product of two linear representations is involved. Let us recall that 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

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

In this section the behavioral approach of [15] is recalled and extended.

Proposition 3.1. *The* parallel composition $G_c \| G$ *of two (max,+) automata*

$$G_c = (Q_c, \alpha_c, t_c, \beta_c),\ G = (Q, \alpha_g, t_g, \beta_g), \quad (7)$$

has the following linear representation (α, μ, β)

$$\begin{aligned} \alpha &= \alpha_c \otimes^t \alpha_g, \\ \forall a \in A_c: \quad \mu(a) &= \mu_c(a) \otimes^t \mu_g(a), \\ \forall a \in A_u: \quad \mu(a) &= E \otimes^t \mu_g(a), \\ \beta &= e_c \otimes^t \beta_g, \end{aligned}$$

in which $e_c = \beta_c$ denotes the column vector of identity elements $e = 0$ of length given by $|Q_c|$, μ_c and μ_g are the morphism matrices corresponding to t_c and t_g, respectively.

Proof. The proof is quite simple and follows from the definition of tensor multiplication and graphical interpretation of morphism matrices. Let us first consider the case $a \in A_c$. It must be shown that

$$\mu(a) = \mu_c(a) \otimes^t \mu_g(a),$$

i.e., that

$$[\mu(a)]_{ik,jl} = [\mu_c(a)]_{ij} \otimes [\mu_g(a)]_{kl}.$$

According to the graphical interpretation of morphism matrix, $[\mu(a)]_{ik,jl}$ is the weight associated to $a-$transition from the state labeled by ik to the state labelled by jl of $G_c \| G$. According to Definition 3.1 it should be equal to the product $\otimes$, *i.e.,* usual sum of $[\mu_c(a)]_{ij}$ the weight of the transition from the state labeled by i to the state labeled by j in the controller G_c and $[\mu_g(a)]_{kl}$ the weight of the transition from the state labeled by k to the state labeled by l of the plant G .
For the case $a \in A_u$, it should be proved that $\mu(a) = E \otimes^t \mu_g(a)$, which according to the definition of tensor product can be rewritten as

$$[\mu(a)]_{ik,jl} = E_{ij} \otimes [\mu_g(a)]_{kl}.$$

The (max,+)-identity matrix is given by

$$E_{ij} = \begin{cases} e, & \text{if } i = j, \\ \varepsilon, & \text{if } i \neq j. \end{cases}$$

The graphical interpretation of $[\mu(a)]_{ik,jl}$ is the weight associated to $a-$transition from the state labelled by ik to the state labeled by jl of $G_c \| G$. According to Definition 3.1 it should be equal either to the weight of the $a-$transition from the state labeled by k to the state labelled by l of G in the case $i = j$, *i.e.,* to $[\mu_g(a)]_{k,l}$ or to ε otherwise (if $i \neq j$).
Concerning the initial delay, it is obvious that $\alpha_{i,j} = (\alpha_c)_i \otimes (\alpha_g)_j$, *i.e.,* $\alpha = \alpha_c \otimes^T \alpha_g$ expresses exactly the fact that the composed state is initial *iff* its components are initial states of C and G.
As discussed following Definition 3.1, it is assumed without loss of generality that all states of the controller are marked. Then β is deduced similarly to α. □

Proposition 3.1 gives the linear representation of the composed system consisting of a controller and a plant, and the behavior can be computed using (4). 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 $\mathbb{R}_{\max}$.

4 Application to supervisory control

Now the parallel composition introduced at section 3 is applied to the supervisory control of (max,+) automata.

A behavioral framework is considered: instead of working with (max,+) automata we work with their behaviors, that is, formal power series from $\overline{\mathbb{R}}_{\max}(A)$ defined by (3) and (4). This is quite natural, because control specifications of supervisory control are typically given by languages which play an analogous role to formal power series of $(\max, +)$ automata. 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 [15]. It consists in separating the logical and the timing aspects of control: first the supremal controllable sublanguage of the specification support is computed and then the timing aspects are considered assuming that all events are controllable (which amounts to $A_c = A$). In this paper we propose a more challenging approach and show how to both handle the logical and timing aspects of the specification, that is, within a single step procedure. To do this, we need a formula for the behavior of the $(\max,+)$ automaton representing the system under supervision, that is, the parallel composition of the controller $(\max,+)$ automaton with the plant $(\max,+)$ automaton. This result is stated in Theorem 4.2 below.

The relationship between tensor product and usual product of matrices, known as the mixed product property, is recalled first (*cf.* [13]).

Property 4.1. *For matrices A, B, C, D of suitable dimensions over a commutative semiring we have:*

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

Theorem 4.2. *The behavior of the parallel composition is the following:*

$$l(G_c \| G)(w) = l(G_c)(P_c(w)) \otimes l(G)(w) = y_c(P_c(w)) \otimes y(w).$$

Proof. We have

$$l(G_c \| G)(w) = \alpha \otimes \mu(w) \otimes \beta.$$

For a given $w = a_1 \dots a_n \in A^*$ we have using morphism property $\mu(w) = \mu(a_1) \dots \mu(a_n)$. From Proposition 3.1 we know that $\mu(a) = \mu_c(a) \otimes^t \mu_g(a)$ for $a \in A_c$ and $\mu(a) = E \otimes^t \mu_g(a)$ for $a \in A_{uc}$. It follows from mixed product property that $\mu(w) = \mu_c(P_c(w)) \otimes^t \mu_g(w)$. Hence,

$$l(G_c \| G)(w) = (\alpha_c \otimes^t \alpha_g) \otimes (\mu_c(P_c(w)) \otimes^t \mu_g(w)) \otimes (\beta_c \otimes^t \beta_g).$$

Finally using once again the mixed product property we obtain:

$$l(G_c \| G)(w) = [\alpha_c \otimes \mu_c(P_c(w)) \otimes \beta_c] \otimes^t [\alpha_g \otimes \mu_g(w) \otimes \beta_g] = l(G_c)(P_c(w)) \otimes l(G)(w).$$

Note that in the tensor product of the last formula scalars are involved. Thus, the tensor product coincides with the scalar multiplication. Hence, $l(G_c \| G)(w) = l(G_c)(P_c(w)) \otimes l(G)(w) = y_c(P_c(w)) \otimes y(w)$ as claimed. □

By comparing the definition of the 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 that is useful for expressing the behavior of the closed-loop system.

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

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

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

The control problem is now described. Let $y_{ref} \in \mathbb{R}_{\max}(A)$ be a specification series and y be the behavior of the uncontrolled plant modeled as a (max,+) automaton, the problem is to find the greatest controller series, denoted y_C such that the closed-loop behavior satisfies $y_C \odot_{A_u} y \preceq y_{ref}$. Having in mind the meaning of the order relation in $\overline{\mathbb{R}}_{\max}(A)$, one can give the following interpretation. Find the greatest y_C, that is, the greatest coefficients $y_C(w)$ for all w, and as a by-product the greatest coefficients $(y_C \odot_{A_u} y)(w)$ with $(y_C \odot_{A_u} y)(w) \preceq y_{ref}(w)$. Note that this inequality has both a logical and a timing aspect. The logical aspect refers to finding the largest support (language) of the controller such that safety inclusion is still guaranted, which amounts to finding the supremal controllable sublanguage of the specification support. The timing aspect 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 is referred to as just-in-time criterion, notably applied for the control of Timed Event Graphs (see for example [3, §5.6],[14]).

Let us introduce the notation

$$H_y^{A_u} : s \mapsto s \odot_{A_u} y \tag{8}$$

for the generalized Hadamard product,otherwise stated,

$$H_y^{A_u}(s) = \bigoplus_{w \in A^*} [s(P_c(w)) \otimes y(w)]\, w.$$

Proposition 2.4 has the following variant in the presence of uncontrollable events ($A_u \neq \emptyset$).

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

$$(H_y^{A_u})^\sharp(s)(w) = \begin{cases} \bigwedge_{u \in P_c^{-1}(w) \cap supp(y)} s(u) \phi y(u), & \text{if } w \in A_c^*, \\ \top, & \text{if } w \notin A_c^*. \end{cases} \tag{9}$$

Proof. The proof goes along the same lines as that of Proposition 2.4. We have

$$(H_y^{A_u} \circ H_y^{A_u \sharp})(s)(w) = H_y^{A_u}[H_y^{A_u \sharp}(s)](w) = H_y^{A_u \sharp}(s)(P_c(w)) \otimes y(w)$$
$$= [\bigwedge_{u \in P_c^{-1} P_c(w) \cap supp(y)} (s(u) \phi y(u))] \otimes y(w),$$

because $P_c(w) \in A_c^*$. Now, it suffices to notice that the set $\{P_c^{-1} P_c(w) \cap supp(y)\}$ either includes w (in case $w \in supp(y)$) or is empty (in case $w \notin supp(y)$). In the former case, we have

$$[\bigwedge_{u \in P_c^{-1} P_c(w) \cap supp(y)} (s(u) \phi y(u))] \preceq s(w) \phi y(w)$$

and $[s(w) \phi y(w)] \otimes y(w) \preceq s(w)$ using $(f.1)$. In the latter case we have $y(w) = \varepsilon$, *i.e.,* in any case $(H_y^{A_u} \circ (H_y^{A_u})^\sharp)(s)(w) = \varepsilon \preceq s(w)$.

In the same manner,

$$(H_y^{A_u \sharp} \circ H_y^{A_u})(s)(w) = H_y^{A_u \sharp}[H_y^{A_u}(s)](w) =$$

$$\begin{cases} \bigwedge_{u \in P_c^{-1}(w) \cap supp(y)} H_y^{A_u}(s)(u) \phi y(u) = \bigwedge_{u \in P_c^{-1}(w) \cap supp(y)} [s(P_c(u)) \otimes y(u) \phi y(u)], & \text{if } w \in A_c^*, \\ \top, & \text{if } w \notin A_c^*. \end{cases}$$

Since we always have $\top \succeq s(w)$ we only care about the first case ($w \in A_c^*$). If $u \in P_c^{-1}(w) \cap supp(y)$, then $P_c(u) \in P_c P_c^{-1}(w) = w$ according to (i) of Lemma 2.5. We then have $[s(P_c(w)) \otimes y(u)] \not\phi y(u) \succeq s(P_c(w)) = s(w)$ thanks to $(f.2)$. The case $\{u \in P_c^{-1}(w) \cap supp(y)\} = \emptyset$ is easy, because the infimum of an empty set is $\top$ and $\top \succeq s(w)$. The conclusion is that $H_y^{A_u}$ is residuated according to Theorem 2.1 with $(H_y^{A_u})^\sharp$ given by the above formula. □

Note that while the value $(H_y^{A_u})^\sharp(s)(w) = \top$ for $w \notin A_c^*$ might seem strange, this value plays no role in the composite (closed-loop) system, because $(H_y^{A_u})^\sharp(s)$ that plays the role of the controller is evaluated only in projected words from A_c^*.

Correctness of this result can also be checked by the following alternative approach.

Using the modified projection P_y (see Definition 2.4), we have in fact $H_y^{A_u} = H_y \circ P_y$, *i.e.*, $\forall s \in \overline{\mathbb{R}}_{\max}(A)$: $H_y^{A_u}(s) = H_y(P_y(s))$. Knowing that ε is absorbing for $\otimes$ 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. *The map P_y is residuated with its residual given by*

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

Proof. We obtain

$$P_y \circ P_y^\sharp(s)(w) = P_y^\sharp(s)(P_c(w)) = \bigwedge_{u \in P_c^{-1} P_c(w) \cap supp(y)} s(u) \preceq s(w),$$

because $P_c(w) \in A_C^*$, hence only the first case in formula for $P_y^\sharp$ can occur.

Similarly, we have

$$P_y^\sharp \circ P_y(s)(w) =$$

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

The arguments are again the same as in the proof of Proposition 4.3. □

Using the formula from residuation theory (*cf.* [3]) $(H_y^{A_u})^\sharp = (H_y \circ P_y)^\sharp = P_y^\sharp \circ H_y^\sharp$, it remains to substitute the formulae for $P_y^\sharp$ (see Proposition 4.4) and $H_y^\sharp$ (see Proposition 2.4). This yields to the same formula as the one obtained in Proposition 4.3.

In the next section Proposition 4.3 will be used in the study of controllability of (max,+) formal power series.

5 Controllability of (max,+) formal power series

In the last section the control problem and its solution based on residuation theory have been formulated within a behavioral framework. The resulting series corresponding to an optimal supervisor can then be realized by a (max,+) automaton, provided it is rational.

In the rest of the paper y denotes the formal power series (behavior) of the uncontrolled plant (system) and y_{ref} the control specification (also called reference series). Similarly as in the classical Ramadge-Wonham (R-W) theory we have y as counterpart of $L(G)$ and y_{ref} as counterpart of the specification language. As in the classical supervisory control theory, not every specification series can be achieved as a closed-loop behavior. Since it is not at first sight clear how to define controllable (max,+) formal power series, controllability is defined by using a property required from a controllable formal power series. Namely, a series is defined to be controllable if it can be exactly achieved by control actions of a suitable supervisor. Formally, we introduce the following concept of controllability within our behavioral framework.

Definition 5.1. *A series $y_{ref} \in \overline{\mathbb{R}}_{\max}(A)$ is controllable with respect to y and A_u if there exists $y_c \in \overline{\mathbb{R}}_{\max}(A)$ such that $y_c \odot_{A_u} y = y_{ref}$, i.e., if $H_y^{A_u}(y_c) = y_{ref}$.*

This concept of controllability may be viewed as an extension of language controllability from R-W theory. The characterization of controllability below follows easily.

Theorem 5.1. *A series $y_{ref} \in \overline{\mathbb{R}}_{\max}(A)$ is controllable with respect to y and A_u iff*

$$y_{ref} = H_y^{A_u} \circ (H_y^{A_u})^\sharp (y_{ref}).$$

Proof. The sufficiency is quite obvious: for any $y_{ref} \in \overline{\mathbb{R}}_{\max}(A)$ such that $y_{ref} = H_y^{A_u} \circ (H_y^{A_u})^\sharp (y_{ref})$, it suffices to take $y_c = (H_y^{A_u})^\sharp (y_{ref})$ and one has $y_{ref} = H_y^{A_u}(y_c)$. For the converse implication, we recall that a residuated mapping satisfies $f \circ f^\sharp \circ f = f$ (see [3, th.4.56]). Therefore, from $y_{ref} = H_y^{A_u}(y_c)$ we get $H_y^{A_u} \circ (H_y^{A_u})^\sharp (y_{ref}) = H_y^{A_u} \circ (H_y^{A_u})^\sharp \circ H_y^{A_u}(y_c) = H_y^{A_u}(y_c) = y_{ref}$, *i.e.,* necessity follows. □

Theorem 5.1 provides a useful characterization of controllable series as those that are fixed points of $H_y^{A_u} \circ (H_y^{A_u})^\sharp$. Note that inequality $H_y^{A_u} \circ (H_y^{A_u})^\sharp (s) \preceq s$ is always satisfied as follows from the very definition of a residuated mapping, *c.f.* Theorem 2.1. Since we have the decomposition $H_y^{A_u} = H_y \circ P_y$ using the standard Hadamard product H_y (corresponding to the absence of uncontrollable events, *i.e.,* $A_c = A$), we obtain that y_{ref} is controllable with respect to y and A_u *iff*

$$y_{ref} = H_y \circ P_y \circ P_y^\sharp \circ H_y^\sharp (y_{ref}).$$

Proposition 4.4 will be helpful in order to obtain the following characterization of controllability that does not refer to the existence of a controller series, but is based purely on the plant and specification series.

Theorem 5.2. *A series $y_{ref} \in \overline{\mathbb{R}}_{\max}(A)$ is controllable with respect to y and A_u iff $\forall w \in A^*$:*

$$y_{ref}(w) \phi y(w) = \bigwedge_{u \in P_c^{-1} P_c(w) \cap supp(y)} y_{ref}(u) \phi y(u).$$

Proof. Controllability of y_{ref} with respect to y and A_u is equivalent to have $\forall w \in A^*$:

$$y_{ref}(w) = H_y \circ P_y \circ P_y^\sharp \circ H_y^\sharp (y_{ref})(w).$$

Then Proposition 4.4 is used, *i.e.,* by mechanically substituting the expressions for $P_y, H_y, P_y^\sharp$, and $H_y^\sharp$ we get: $\forall w \in A^*$: $y_{ref}(w) = \{\bigwedge_{u \in P_c^{-1} P_c(w) \cap supp(y)} y_{ref}(u) \phi y(u)\} \otimes y(w)$, whence

$$y_{ref}(w) \phi y(w) = \bigwedge_{u \in P_c^{-1} P_c(w) \cap supp(y)} y_{ref}(u) \phi y(u)$$

(for scalars $a, b \in \overline{\mathbb{R}}_{\max}$ we have $a = b \otimes x \Leftrightarrow a = b + x \Leftrightarrow a - b = x \Leftrightarrow a \phi b = x$). □

The claim of Theorem 5.2 can be reformulated as follows: $\forall w \in A^*$ and $\forall u \in P_c^{-1}P_c(w) \cap supp(y)$

$$y_{ref}(u) \not{\phi} y(u) \succeq y_{ref}(w) \not{\phi} y(w)$$

Otherwise stated, we must have equality for any $w \in supp(y)$, because clearly such w belongs to $\{u \in P_c^{-1}P_c(w) \cap supp(y)\}$. Indeed, if $w \notin supp(y)$ then we obtain $\top$ on the right (because $a \not{\phi} \varepsilon$ for any $a \in \overline{\mathbb{R}}_{\max}$, included $a = \varepsilon$) and since $u \in supp(y)$ we obtain $\top$ on the left as well. Formally, the following corollary holds true.

Corollary 5.3. *A series $y_{ref} \in \overline{\mathbb{R}}_{\max}(A)$ is controllable with respect to y and A_u iff $\forall w \in supp(y)$ and $\forall u \in P_c^{-1}P_c(w) \cap supp(y)$, we have*

$$y_{ref}(u) \not{\phi} y(u) = y_{ref}(w) \not{\phi} y(w).$$

5.1 Logical and timing aspects of controllability

Note that in the characterization of controllability above both logical and timing aspects of controllability are included.

Let us first discuss the timing aspects of controllability. Since y_{ref} as well as y are scalar series, *i.e.,* all coefficients are numbers (including ε), one can reformulate controllability stated in Corollary 5.3 as

$$\forall w \in supp(y) \text{ and } \forall u \in P_c^{-1}P_c(w) \cap supp(y) \;:$$

$$y_{ref}(w) \not{\phi} y_{ref}(u) = y(w) \not{\phi} y(u).$$

Note that $u \in P_c^{-1}P_c(w)$ just means that u and w differ only by uncontrollable events. Now, if $w \succeq u$, then the formula expresses the requirement that given a time delay between the occurrence of strings u and w within the system ($y(w) \not{\phi} y(u)$), the same delay between the strings u and w must be prescribed by the specification series ($y_{ref}(w) \not{\phi} y_{ref}(u)$). This is a very natural and intuitive requirement, because the intermediate uncontrollable events (that make the difference between those strings : $P_c(u) = P_c(w)$) cannot be delayed by any controller automaton.

Let us now discuss the logical aspects of controllability and compare it with controllability of languages. To do this, we first define a projection $\tilde{P}_c : A^* \to A^*$ that removes uncontrollable strings (if any) at the end of words. Thus, $\tilde{P}_c(w) = v$ if $w = vu,\ u \in A_u^*$ and $last(v) \in A_c$, where $last(v)$ denotes the last letter of the word v.

Proposition 5.4. *A prefix closed language K is controllable with respect to L and A_u iff $\tilde{P}_c^{-1}\tilde{P}_c(K) \cap L \subseteq K$.*

Proof. Follows from the definition of controllability of a language [23] that can be equivalently given using strings of A_u^* instead of events of A_u, *i.e.,* $KA_u^* \cap L \subseteq K$. Indeed, let $KA_u^* \cap L \subseteq K$ and $s \in \tilde{P}_c^{-1}\tilde{P}_c(K) \cap L$, then $s \in L$ and $\tilde{P}_c(s) = \tilde{P}_c(t)$ for some $t \in K$. From definition of $\tilde{P}_c$ it follows that either $s = tu$ with $u \in A_u^*$ or $t = su$ with $u \in A_u^*$. In the first case we get $s = tu \in K$ by controllability of K and in the second case $t = su \in K$ implies $s \in K$, because K is prefix closed.

Similarly, if $\tilde{P}_c^{-1}\tilde{P}_c(K) \cap L \subseteq K$ and $s = tu \in L$ with $t \in K$ and $u \in A_u^*$, then $s \in \tilde{P}_c^{-1}\tilde{P}_c(K) \cap L$, *i.e.,* $s \in K$. □

Now we return to the characterization of controllability of series and we extract logical aspects of it to compare with controllability of languages. In this respect, as mentioned in Remark 2.6 it is sufficient to consider the support of series (*i.e.,* series with Boolean coefficients) instead of series having coefficients in $\overline{\mathbb{R}}_{\max}$ (any coefficient different from ε, including $\top$ becomes the unit element e). The series y_{ref} plays the role of the specification language K, *i.e.,* $y_{ref}(w) = e$ means that $w \in K$ and similarly $y(w) = e$ means that $w \in L$. One can notably check that Proposition 5.4 implies characterization of controllability stated in Corollary 5.3. To do this, let us consider a controllable prefix closed language K and $w \in K$ (*i.e.,* $y_{ref}(w) = e$), from Proposition 5.4 we have $\tilde{P}_c^{-1}\tilde{P}_c(K) \cap L \subseteq K$, and in particular, $\forall u \in \tilde{P}_c^{-1}\tilde{P}_c(w) \cap L$, *i.e.,* $y(w) = e$ and $u \in \tilde{P}_c^{-1}\tilde{P}_c(w)$ which implies $u \in P_c^{-1}P_c(w)$, we have $u \in K$, *i.e.,* $y_{ref}(u) = e$. Then $\forall w \in supp(y)$, *i.e.,* $y(w) = e$, we have the condition of Corollary 5.3, that is,

$$y_{ref}(u) \phi y(u) = y_{ref}(w) \phi y(w) = e \phi e = e.$$

The converse implication is not true. More precisely, in the converse reasoning, one cannot argue that $u \in P_c^{-1}P_c(w)$ implies $u \in \tilde{P}_c^{-1}\tilde{P}_c(w)$.

This makes a connection between (max,+) controllability and logical controllability. More precisely, this means that our original notion of controllability for formal power series (with P_c instead of $\tilde{P}_c$) is stronger in its logical aspect than classical R-W controllability of languages. Since there is no notion of prefix closed behaviors for formal power series, the control problem, that has been formulated for formal power series that are counterparts of marked languages, is more restrictive (*cf.* for languages inclusion of marked languages implies inclusion of prefix closed languages if the systems are nonblocking). Hence, controllability needs to be stronger.

5.2 Supremal controllable behaviors

If a specification series is not controllable, a natural question is to find an approximation, in particular a smaller series, that is controllable.

Let us first notice that $H_y^{A_u}$ and $(H_y^{A_u})^\sharp$ are isotone mappings. The following result holds.

Proposition 5.5. *$H_y^{A_u} \circ (H_y^{A_u})^\sharp(y_{ref})$ is the greatest controllable (max,+) series with respect to y and A_u smaller or equal to y_{ref}.*

Proof. Controllability follows from the very definition, *cf.* proof of Theorem 5.1. It remains to show the supremality of $H_y^{A_u} \circ (H_y^{A_u})^\sharp(y_{ref})$ among all controllable series that are less or equal to y_{ref}. Let $\bar{y}$ be controllable with respect to y and A_u and $\bar{y} \preceq y_{ref}$. By isotony of $H_y^{A_u}$ and $(H_y^{A_u})^\sharp$ one obtains $H_y^{A_u} \circ (H_y^{A_u})^\sharp(\bar{y}) \preceq H_y^{A_u} \circ (H_y^{A_u})^\sharp(y_{ref})$, and since $H_y^{A_u} \circ (H_y^{A_u})^\sharp(\bar{y}) = \bar{y}$, we have $\bar{y} \preceq H_y^{A_u} \circ (H_y^{A_u})^\sharp(y_{ref})$ which shows the supremality of $H_y^{A_u} \circ (H_y^{A_u})^\sharp(y_{ref})$. ☐

Remark 5.6. *There is an analogy with the classical supervisory control theory. 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 residuated mapping $(H_y^{A_u}) \circ (H_y^{A_u})^\sharp(s)$ 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. Firstly, $H_y^{A_u}(s)$ plays the role of closed-loop behavior of the controlled system. In classical supervisory control it corresponds to the infimal controllable superlanguage. However, in our case, where timing aspect of control is defined by adding delay, i.e., (max,+) multiplication, we cannot expect that the supremal controllable subseries of a controllable series is this series itself. Therefore it is not $H_y^{A_u})^\sharp(s)$, but $(H_y^{A_u}) \circ (H_y^{A_u})^\sharp(s)$ that is the formal power series counterpart of the supremal controllable sublanguage of s.

The last proposition can then be viewed as a generalization of the formula for $\sup \mathrm{C}$ *operator from Ramadge-Wonham theory.*

Corollary 5.7. *If* y_{ref} *is controllable with respect to* y *and* A_u *then the optimal controller series is simply given by*

$$y_c(w) = \begin{cases} y_{ref}(w) \not\phi y(w) & , \text{ if } w \in A_c^*, \\ \top & , \text{ if } w \notin A_c^*. \end{cases}$$

This is similar to classical supervisory control, where controller is simply given by the intersection of the plant and the specification languages if the specification is controllable.

6 Example

The aim of this section is to illustrate various concepts and results introduced in this paper. To do this a simple example is preferred to a realistic case study.
A manufacturing system modeled by a (max,+) automaton G displayed on figure 1.(a) is considered. The three distinct tasks, labeled 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, $abcb$, $abcbc$, $\ldots$. The linear representation of G (see section 2.3) is given by

$$\alpha = \begin{pmatrix} e & \varepsilon & \varepsilon \end{pmatrix}, \ \beta = \begin{pmatrix} \varepsilon \\ e \\ e \end{pmatrix},$$

$$\mu(a) = \begin{pmatrix} \varepsilon & 3 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix}, \ \mu(b) = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 4 \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix}, \ \mu(c) = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 5 & \varepsilon \end{pmatrix}.$$

The behavior of G can be deduced using (4), that is by computing

$$y = l(G) = \alpha \otimes \mu^* \otimes \beta = \alpha \otimes (\mu(a)a \oplus \mu(b)b \oplus \mu(c)c)^* \otimes \beta = \alpha \otimes \begin{pmatrix} \varepsilon & 3a & \varepsilon \\ \varepsilon & \varepsilon & 4b \\ \varepsilon & 5c & \varepsilon \end{pmatrix}^* \otimes \beta.$$

We obtain the following series in $\overline{\mathbb{R}}_{\max}(A)$:

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

For instance, $y(ab) = 7$ means that the sequence ab will be completed at the 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 which means that this task starts as soon as it can be performed. Denoting $A = \{a, b, c\}$ the set of events (alphabet), we have $A_c = \{a, c\}$ and $A_u = \{b\}$.
We would like that the system behaves 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 on figure 1.(b).

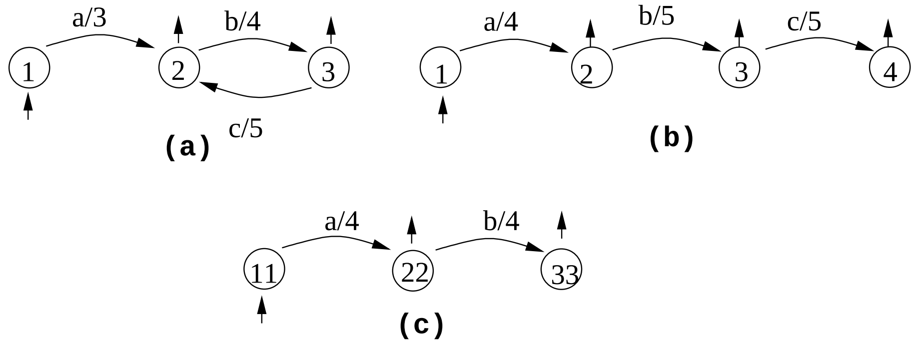

Figure 1: G **(a)**, G_{ref} **(b)**, G_s **(c)**

The series y_{ref} is not controllable with respect to y and A_u since for $w = abc$ we have $P_c^{-1}P_c(w) = b^*ab^*cb^*$, that is, $P_c^{-1}P_c(w) \cap supp(y) = \{abc,\, abcb\}$ and

$$\begin{array}{rcl} y_{ref}(abc)\not\phi y(abc) & & y_{ref}(abc)\not\phi y(abc) \wedge y_{ref}(abcb)\not\phi y(abcb) = \\ = 14\not\phi 12 & & \min(14\not\phi 12, \varepsilon\not\phi 16) = \\ = 2 & \neq & \varepsilon. \end{array}$$

The condition stated in Theorem 5.2 is then violated. This is due to the uncontrollability of event b, which implies that word $abcb$ ($\in supp(y)$) cannot be prevented from happening if abc ($\in supp(y_{ref})$)) is enabled.
In addition, for $w = ab$ we have $P_c^{-1}P_c(w) = b^*ab^*$, that is, $P_c^{-1}P_c(w) \cap supp(y) = \{a,\, ab\}$ and

$$\begin{array}{rcl} y_{ref}(ab)\not\phi y(ab) & & y_{ref}(a)\not\phi y(a) \wedge y_{ref}(ab)\not\phi y(ab) = \\ = 9\not\phi 7 & & \min(4\not\phi 3, 9\not\phi 7) = \\ = 2 & \neq & 1. \end{array}$$

The condition of Theorem 5.2 is again not satisfied. In that case, this is due to the fact that if event a has been completed at 4, then event b cannot be delayed (since it is uncontrollable) so that sequence ab would be completed at 9.

Proposition 5.5 can be used to compute the greatest controllable series with respect to y and A_u smaller or equal to y_{ref}. We obtain

$$H_y^{A_u} \circ (H_y^{A_u})^\sharp(4a \oplus 9ab \oplus 14abc) = 4a \oplus 8ab.$$

In fact, we have using (9) for $y_c^{opt} = (H_y^{A_u})^\sharp$

$$(H_y^{A_u})^\sharp(ab) \;=\; \top \quad , \quad (H_y^{A_u})^\sharp(abc) \;=\; \top \quad (\text{since } ab \notin A_c^* \text{ and } abc \notin A_c^*)$$

and

$$\begin{array}{rcl} (H_y^{A_u})^\sharp(a) & = & y_{ref}(a)\not\phi y(a) \wedge y_{ref}(ab)\not\phi y(ab) \\ & = & 1, \end{array}$$

$$\begin{array}{rcl} (H_y^{A_u})^\sharp(ac) & = & y_{ref}(abc)\not\phi y(abc) \wedge y_{ref}(abcb)\not\phi y(abcb) \\ & = & \varepsilon. \end{array}$$

Since there is no $u \in P_c^{-1}(\varepsilon) \cap supp(y)$, *i.e.,* $A_{uc}^* \cap supp(y) = \emptyset$ and the infimum of empty set is $\top$, we have $(H_y^{A_u})^\sharp(\varepsilon) = \top$. Hence, the controller is

$$y_c^{opt}(\varepsilon) = (H_y^{A_u})^\sharp(\varepsilon) = \top,\ y_c^{opt}(a) = (H_y^{A_u})^\sharp(a) = 1, \text{ and}$$

$$y_c^{opt}(w) = (H_y^{A_u})^\sharp(w) = \varepsilon \text{ for } w \in A_c^*,\ w \notin \{\varepsilon, a\}$$

and $y_c^{opt}(w) = \top$ for $w \notin A_c^*$.

Also, this leads to

$$\begin{aligned} H_y^{A_u} \circ (H_y^{A_u})^\sharp(abc) &= (H_y^{A_u})^\sharp(P_c(abc)) \otimes y(abc) \\ &= (H_y^{A_u})^\sharp(ac) \otimes y(abc) \\ &= \varepsilon \otimes 12 = \varepsilon, \end{aligned}$$

$$\begin{aligned} H_y^{A_u} \circ (H_y^{A_u})^\sharp(ab) &= (H_y^{A_u})^\sharp(P_c(ab)) \otimes y(ab) \\ &= (H_y^{A_u})^\sharp(a) \otimes y(ab) \\ &= 1 \otimes 7 = 8, \end{aligned}$$

$$\begin{aligned} H_y^{A_u} \circ (H_y^{A_u})^\sharp(a) &= (H_y^{A_u})^\sharp(P_c(a)) \otimes y(a) \\ &= (H_y^{A_u})^\sharp(a) \otimes y(a) \\ &= 1 \otimes 3 = 4. \end{aligned}$$

The behavior of the system under control is $y_s = y_c^{opt} \odot y = \bigoplus_{w \in A^*} \left(y_c^{opt}(P_c(w)) \otimes y(w) \right) w = 4a \oplus 8ab$. A (max,+) automation G_s which realizes y_s is displayed in figure 1.(c). The role of the controller in this example is to delay the first event (a) by 1 unit and to disable the event c after the string ab has been generated by the system.

7 Rationality and decidability issues

This section is dedicated to rationality and decidability issues related to (max,+) formal power series.

Let us recall that a series is (max,+)- rational (respectively (min,+)- rational) if it is in the rational closure of series with finite supports, *i.e.,* if it can be formed from polynomial series (*i.e.,* those with finite support) by rational operation $\oplus$ (corresponding to $\max$, respectively to $\min$), $\otimes$, and the Kleene star.

Interestingly, it has been shown in [22] that the class of formal power series that are at the same time (max,+) and (min,+) rational coincides with the so called *unambiguous series*. For these families of series, the equality (and inequality) of series is proved to be decidable. Unambiguous series are recognized by unambiguous automata, that is, automata in which there is at most one successful path labeled by w for every word w. If we confine ourselves to this class of automata (series), which roughly consist to consider deterministic automata, there is no problem with decidability of inequalities. This will be adopted in our future extensions of the proposed framework, in particular in decentralized control of synchronous products of deterministic (max,+) automata. Synchronous product of (max,+) automata will be defined as a multi-event (max,+) automaton, more precisely (max,+) automaton defined over local sequences based alphabet. This way a use of nondeterminism to model temporal aspects of concurrency (as in heap automata from [11]) may be avoided.

Another important question is whether/when the resulting controller is rational. This amounts to study the rationality of residuated mapping of Hadamard product. The Hadamard product of two rational power series is still a rational power series when the underlying semiring of coefficients is a commutative semiring. The situation is more complicated when the Hadamard inverse (quotient) is considered. It turns out to be a difficult problem, but the results of [22] are again helpful in this respect. Actually, the residuated mapping of a Hadamard product can be formulated in terms of Hadamard product using 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 $H_y^\sharp(s)(w) = s(w) \not{/} y(w) = s \odot Cy$. 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 [22] that for a formal power series $s \in \mathbb{R}_{\max}(A)$ we have $C(s) \in \mathbb{R}_{\max}(A)$ *iff* s is unambiguous. Moreover we recall that it is proven therein that $s \in \mathbb{R}_{\max}(A)$ is unambiguous *iff* it is at the same time (max,+) and (min,+) rational. It is then clear in view of our formula (9) that if the series $y \in \mathbb{R}_{\max}(A)$ (corresponding to the uncontrolled plant) is not at the same time (max,+) and (min,+) rational, the residuated mapping of Hadamard product can neither be rational.

Acknowledgement

This work was supported by the Academy of Sciences of the Czech Republic, Inst. Research Plan No. AV0Z10190503 and by EU.ICT project DISC.

8 Conclusion

We have presented a control mechanism for (max,+) automata based on the tensor product of their linear representations, *i.e.,* the 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 an approach based on a generalized version of Hadamard product and on direct application of residuation theory is developed. This way both logical and timing aspects of supervisory control are handled at the same time.

The proposed solution to a control problem for (max,+) automata is used in the study of controllability. Controllability of (max,+) formal power series is investigated using residuation theory applied to a generalized Hadamard product of formal power series. Both logical and timing aspects of controllability are characterized within a single formula. Supremal controllable behaviors have been studied and formulae are proposed for supremal controllable series. In a future investigation it would be nice to handle unobservable events and to develop decentralized and modular control of concurrent (max,+) automata. It would be also interesting to specify results to $(\max, +)$ automata with only positive transition values, that is the more realistic case where durations associated to events are all positive or zero (in case of instantaneous actions). Moreover, the algebraic approach could be extended to more general classes of timed automata.

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] F. Baccelli, G. Cohen, G.J. Olsder and J.-P. Quadrat (1992). *Synchronization and Linearity. An Algebra for Discrete Event Systems.* New York, Wiley.

[4] A. Benveniste, C. Jard, and S. Gaubert. *Algebraic Techniques for Timed Systems.* In Proceedings of International Conference on Concurrency Theory (CONCUR'98), 1998.

[5] A. Benveniste, S. Gaubert, and C. Jard. *Monotone Rational Series and Max-plus Algebraic Models of Real-time Systems.* In Proceedings of 4th Workshop on Discrete Event Systems (WODES'98), Cagliari, Italy, August 1998.

[6] J. Berstel and C. Reutenauer. *Rational Series and their Languages.* Berlin, Springer Verlag, 1988.

[7] T.S. Blyth and M.F. Janowitz (1972). *Residuation theory.* Oxford, Pergamon Press.

[8] B.A. Brandin and W.M. Wonham. *Supervisory Control of Timed Discrete Event Systems*, IEEE Transations on Automatic Control 39 (2), pp. 329-351, 1994.

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

[10] S. Gaubert. *Performance Evaluation of (max,+) Automata*, IEEE Transaction on Automatic Control, 40(12), pp. 2014-2025, 1995.

[11] 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.

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

[13] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press (United Kingdom), 1994.

[14] L. Houssin, S. Lahaye, J.-L. Boimond, *Just in Time Control of constrained (max,+)-Linear Systems*. Discrete Event Dynamic Systems: Theory and Applications, 17(2), pp. 159-178, 2007.

[15] 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.

[16] J. Komenda, S. Lahaye, and J.-L. Boimond. Supervisory Control of (max,+) Automata: A single Step Approach. European Control Conference (ECC) 2009, Budapest, Hungary.

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

[18] 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.

[19] R. Kumar and M. Heymann. *Masked Prioritized Synchronization for Interaction and Control of Discrete Event Systems*. IEEE Transactions on Automatic Control, pages 1970-1982, volume 45, number 11, Nov. 2000.

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

[21] S. Lombardy and J. Sakarovitch. *Derivatives of Rational Expressions with Multiplicity.* Theoretical Computer Science 332, pp. 141-177, 2004.

[22] 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.

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

[24] A.J. Van der Poorten, *Solution de la conjecture de Pisot sur le quotient de Hadamard de deux fractions rationnelles*, C.R. Acad. Sci. Paris 306, série I, 1988.

[25] H. Wong-Toi, G. Hoffmann, The Control of Dense Real-Time Discrete Event Systems, Technical Report STA-CS-92-1411, Stanford University, 1995.