
Le produit synchrone des automates $(\max,+)$

Jan Komenda* — Sébastien Lahaye** — Jean-Louis Boimond**

* *Institut de Mathématiques, Académie Tchèque des Sciences, Žitkova 22, 616 62 Brno, République Tchèque, komenda@ipm.cz*

** *LISA, 62 avenue Notre-Dame du Lac, 49000 Angers, France
{sebastien.lahaye,jean-louis.boimond}@univ-angers.fr*

RÉSUMÉ. Une extension des automates $(\max,+)$ est étudiée dans le but de modéliser le parallélisme (occurrence simultanée d'événements). Pour cela, on introduit une composition synchrone des automates $(\max,+)$ vus comme des automates temporisés. Ceci nous amène à introduire des automates $(\max,+)$ avec multi-événements qui correspondent à une classe des automates temporisés avec plusieurs horloges. Nous obtenons la formule pour le comportement de produit synchrone d'automates $(\max,+)$ et montrons que dans le cas général il n'est pas possible de définir le produit synchrone des comportements (séries formelles) sans prendre en compte leurs représentations par automates $(\max,+)$.

ABSTRACT. Synchronous products of $(\max,+)$ automata are introduced as multi-event $(\max,+)$ automata (which correspond to a class of timed automata with several clocks) and their behavior is studied. A formula for the behavior of synchronous product is obtained.

MOTS-CLÉS : automates $(\max,+)$, produit synchrone, parallélisme

KEYWORDS: $(\max,+)$ automata, synchronous product, parallelism

1. Introduction

Les techniques de commande supervisée des systèmes à événements discrets, proposées par Ramadge et Wonham (Ramadge *et al.*, 1989), concernent des aspects logiques. La nécessité de considérer des contraintes temporelles a conduit à étendre ces techniques au cas de systèmes temporisés. On peut notamment retenir une extension aux cas systèmes de temps-réel, basée sur des abstractions d'automates temporisés en automates booléens (logiques), appelés également automates des régions (Wong-Toi *et al.*, 1995).

Une classe de systèmes à événements discrets temporisés peut être modélisée par des automates $(\max, +)$ (Gaubert, 1995). Dans le but de maîtriser leur comportement, une méthodologie de commande des automates $(\max, +)$ a été développée dans (Komenda *et al.*, 2009). Elle repose sur la composition parallèle du contrôleur avec le système. Cette composition parallèle n'est pas la composition synchrone, mais spécifie l'interaction du contrôleur avec le système en présence d'événements incontrôlables. Elle se réduit à la composition classique des automates à multiplicités (Arnold, 1994) (à valeurs dans le semi-anneau $(\mathbb{R} \cup \{-\infty\}, \max, +)$) dans le cas où tous les événements sont contrôlables. En termes de représentations linéaires des automates $(\max, +)$, la composition parallèle correspond au produit tensoriel des matrices sur le semi-anneau $(\mathbb{R} \cup \{-\infty\}, \max, +)$, où la matrice de morphisme du contrôleur est remplacée par la matrice d'identité pour les événements incontrôlables. Le contrôle optimal (le moins restrictif) des automates $(\max, +)$ vis-à-vis du critère juste-à-temps a été proposé et est rappelé ici. Il porte sur l'application de la théorie de la résiduation à une extension du produit de Hadamard des séries formelles.

Dans ce papier, nous étudions une extension des automates $(\max, +)$ pour la modélisation d'une classe plus générale d'automates temporisés. Plus précisément, on introduit une composition synchrone qui conduit aux automates $(\max, +)$ sur un alphabet dit multi-événements (composé des sous-ensembles d'événements).

Le papier est organisé comme suit. Dans la section suivante, on rappelle les outils algébriques importants pour notre approche. La section 3 est dédiée aux automates $(\max, +)$ et à leur composition parallèle pour la commande. Nous présentons dans la Section 4 l'extension de notre approche au produit synchrone des automates $(\max, +)$.

2. Préliminaires

Dans cette section, on fait un rappel succinct sur les semi-anneaux idempotents (voir (Baccelli *et al.*, 1992) pour une présentation exhaustive).

2.1. Semi-anneau idempotent

Définition 2.1. *Un semi-anneau idempotent (aussi appelé dioïde) est un ensemble \mathcal{D} muni de deux opérations binaires : l'addition et la multiplication. L'addition \oplus est*

commutative, associative, et admet un élément nul, noté ε , (i.e., $\varepsilon \oplus a = a$ pour tout $a \in \mathcal{D}$). De plus, l'opération \oplus est idempotente, (i.e., $a \oplus a = a$ pour tout $a \in \mathcal{D}$). La multiplication \otimes est associative, possède l'élément neutre e (i.e., $e \otimes a = a \otimes e = a$ pour tout $a \in \mathcal{D}$), sachant que \otimes est distributif par rapport à \oplus . De plus, ε est absorbant pour \otimes , i.e., $\forall a \in \mathcal{D} : a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$.

Dans tout dioïde, l'ordre naturel \preceq est défini par : $a \preceq b \Leftrightarrow a \oplus b = b$. Un dioïde \mathcal{D} est complet si tout sous ensemble A de \mathcal{D} admet une plus petite borne supérieure, notée $\bigoplus_{x \in A} x$, et si \otimes distribue sur les sommes infinies. En particulier, $T = \bigoplus_{x \in \mathcal{D}} x$ est le plus grand élément de \mathcal{D} . Dans un dioïde complet, la plus grande borne inférieure \wedge existe toujours : $a \wedge b = \bigoplus_{x \preceq a, x \preceq b} x$.

Comme exemple de dioïdes, on peut citer ceux portant sur des nombres, des langages ou des séries formelles. En particulier, le dioïde $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +)$ manipule des nombres réels où le maximum joue le rôle de l'addition, notée $\oplus : a \oplus b = \max(a, b)$, et l'addition classique celui de la multiplication, notée $a \otimes b$ (ou plus simplement lorsque cela ne prête pas à confusion ab). Le dioïde \mathbb{R}_{\max} , complété par l'élément $T = +\infty$, est noté $\overline{\mathbb{R}}_{\max}$, il est complet, sachant que $T \otimes \varepsilon = \varepsilon \otimes T = \varepsilon$. Les opérations matricielles sont définies comme dans l'algèbre linéaire classique. La matrice d'identité de $\mathbb{R}_{\max}^{n \times n}$ est notée E , i.e.

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

Dans un dioïde complet, on définit l'étoile de Kleene d'un élément a comme suit :

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

où par convention $a^0 = e$ et $a^n = a^{n-1} \otimes a$.

2.2. Dioïde des séries formelles $\overline{\mathbb{R}}_{\max}(A)$

Les langages formels sur un alphabet A sont des sous ensembles du monoïde libre A^* lequel est constitué des séquences finies de lettres (mots) de A . L'ensemble des langages formels doté de l'addition (correspondant à l'union des langages) et de la multiplication (correspondant à la concaténation des langages) est un dioïde, noté par $(Pwr(A^*), \cup, \cdot)$. Le langage nul est noté $0 = \{\}$, le langage unité est noté $1 = \{\varepsilon\}$ où ε est le mot vide. Un mot $u = u_1 \dots u_k \in A^*$ est appelé un sous mot de $v \in A^*$ s'il existe une factorisation $v = v_1 u_1 v_2 \dots v_k u_k v_{k+1}$ avec $v_i \in A^*$, $i = 1, \dots, k+1$. L'ordre induit des sous mots sur A^* est $u \preceq v$ si u est un sous mot de $v \in A^*$. Le dioïde des séries formelles à variables dans A et à coefficients dans \mathbb{R}_{\max} , notée $\mathbb{R}_{\max}(A)$, est muni de l'addition mot à mot et d'une multiplication sous forme de convolution. Aussi, pour $s = \bigoplus_{w \in A^*} s(w)w$ et $s' = \bigoplus_{w \in A^*} s'(w)w$ dans $\mathbb{R}_{\max}(A)$, on a :

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

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

Le dioïde des séries formelles est complet si les coefficients sont dans $\overline{\mathbb{R}}_{\max}$.

Notons que pour $s, s' \in \mathbb{R}_{\max}(A)$, $s \preceq s'$ (l'ordre naturel sur $\mathbb{R}_{\max}(A)$) correspond à $s(w) \leq s'(w)$ pour tout $w \in A^*$. Le langage $\text{supp}(s) = \{w \in A^* : s(w) \neq -\infty\}$ est appelé le support de la série s . Une série formelle est reconnue par un automate fini (max,+)*ssi* elle est rationnelle, *i.e.*, elle est formée par des opérations rationnelles à partir de séries polynomiales (celles avec un support fini).

3. Automates (max,+) et leur comportement

Les automates à multiplicités dans le semi-anneau \mathbb{R}_{\max} , encore appelés automates (max,+), ont été introduits par S. Gaubert dans (Gaubert, 1995) et appliqués à l'analyse de performances des systèmes à événements discrets. Ils sont à même de modéliser des phénomènes de synchronisation et de partage des ressources, ils généralisent à la fois les automates logiques et les systèmes (max,+)-linéaires. Les multiplicités peuvent avoir différentes interprétations (telle celle d'un coût pour des automates (min,+)), mais le plus souvent elles représentent le temps, *i.e.* la durée des transitions. Les définitions et résultats de base sur les automates sont présentés dans (?) et (Gaubert *et al.*, 1999).

Définition 3.1. *Un automate (max,+) défini sur un alphabet A est un quadruplet $G = (Q, A, \alpha, t, \beta)$, où Q est l'ensemble fini d'états, A est l'ensemble fini d'événements (encore appelé alphabet), $t : Q \times A \times Q \rightarrow \mathbb{R}_{\max}$ est la fonction de transition, $\alpha : Q \rightarrow \mathbb{R}_{\max}$ et $\beta : Q \rightarrow \mathbb{R}_{\max}$ correspondant respectivement aux délais initiaux et finaux.*

La fonction de transition associe à un état $q \in Q$, un événement $a \in A$ et un nouvel état $q' \in Q$, une valeur de sortie $t(q, a, q') \in \mathbb{R}$ correspondant à la durée de transition a de q vers q' , sachant que $t(q, a, q') = \varepsilon$ s'il n'y a pas de transition de q vers q' avec le label a . La fonction de transition de la définition 3.1 est en général non déterministe, car pour $q \in Q$ et $a \in A$ donnés, il peut exister plusieurs $q' \in Q$ tels que $t(q, a, q') \neq \varepsilon$. Un automate (max,+) avec la fonction de transition non déterministe est appelé non déterministe.

On dit que $q \in Q$ est un état initial (resp. final) si $\alpha(q) \neq \varepsilon$ (resp. $\beta(q) \neq \varepsilon$).

Un automate (max,+) est souvent défini par un triplet (appelé représentation linéaire) (α, μ, β) où $\alpha \in \mathbb{R}_{\max}^{1 \times |Q|}$, $\beta \in \mathbb{R}_{\max}^{|Q| \times 1}$ et μ est le morphisme :

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

Notons que la matrice de morphisme d'un automate (max,+) peut aussi être considérée comme un élément de $\mathbb{R}_{\max}(A)^{|Q| \times |Q|}$, c'est-à-dire $\mu = \bigoplus_{w \in A^*} \mu(w)w$ en

étendant la définition de μ de A aux mots $w \in A^*$ à l'aide de la propriété de morphisme :

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

Notons que μ admet la propriété importante d'être finiment générée parce qu'elle est complètement déterminée par ses valeurs sur A . Nous avons en effet $\mu^* = (\bigoplus_{a \in A} \mu(a)a)^*$. Etant intéressé par les comportements des automates (max,+), définis par $l = \alpha \mu^* \beta$ (voir ci-dessous), nous écrivons par abus de notation $\mu = \bigoplus_{a \in A} \mu(a)a$.

Le comportement d'un automate (max,+) $G = (Q, A, \alpha, t, \beta)$ est donné par la série formelle $l(G) : A^* \rightarrow \mathbb{R}_{\max}$ suivante : pour $w = a_1 \dots a_n \in A^*$

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

q_0 (resp. q_n) est un état initial (resp. final). Autrement dit, $l(G)(w)$ est le poids maximal de tous les chemins valués par w menant d'un état initial vers un état final.

REMARQUE. — La série formelle $l(G) : A^* \rightarrow \mathbb{R}_{\max}$ est un dateur généralisé (Gaubert, 1995), où $l(G)(w)$ représente la date à laquelle la séquence des événements (les tâches) w a été achevée avec la convention $l(G)(w) = -\infty = \varepsilon$ si w n'a pas lieu. Par une spécialisation aux séries booléennes, c'est-à-dire à coefficients dans $\{\varepsilon, e\}$, nous obtenons la notion classique de langage dans la théorie de Ramadge et Wonham, *i.e.*, $l(G)(w) \neq \varepsilon$ si w est un comportement admissible de l'automate. Par conséquent, il suffit de travailler avec les supports des séries qui correspondent aux comportements des automates (max,+) afin d'étudier les aspects logiques de leurs commandes.

De la même manière que les graphes d'événements temporisés admettent une description par des équations de type point fixe linéaires dans le dioïde des séries formelles $Z_{\max}(\gamma)$, (Baccelli *et al.*, 1992, §5.3), tout automate (max,+) peut être décrit linéairement dans le dioïde $\mathbb{R}_{\max}(A)$ des séries formelles comme suit :

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

où $\mu = \bigoplus_{a \in A} \mu(a)a \in \mathbb{R}_{\max}(A)^{|Q| \times |Q|}$ est la matrice de morphisme. Rappelons que la plus petite solution des équations précédentes est donnée par :

$$y = l(G) = \alpha \otimes \mu^* \otimes \beta. \quad [2]$$

Ceci est en accord avec la formule 1 exprimant le comportement d'un automate (max,+), car en termes des matrices la formule 1 s'écrit comme suit :

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

Récemment nous avons proposé une extension des techniques de commande supervisée aux automates $(\max,+)$, *c.f.* (Komenda *et al.*, 2009). Nous avons considéré le produit parallèle de l'automate $(\max,+)$ représentant le système avec celui qui représente le contrôleur (superviseur). En termes de représentations linéaires des automates $(\max,+)$, la composition parallèle correspond au produit tensoriel des matrices sur le semi-anneau $(\mathbb{R} \cup \{-\infty\}, \max, +)$, où la matrice de morphisme du contrôleur est remplacée par la matrice d'identité pour les événements incontrôlables. Le contrôle optimal (le moins restrictif) des automates $(\max,+)$ vis-à-vis du critère juste-à-temps a été proposé. Cependant, l'opération du produit parallèle de (Komenda *et al.*, 2009) ne peut pas représenter le produit synchrone des automates $(\max,+)$ vus comme automates temporisés spéciaux. En effet, dans le produit parallèle l'événement autorisé par le superviseur est retardé dans le système composé, plus précisément sa durée dans le système composé est la somme des durées de cet événement dans le système et dans le superviseur. Ceci est adéquat pour représenter l'aspect temporel de supervision (retarder les événements contrôlables), mais est loin d'exprimer la composition synchrone, où la durée d'un événement partagé dans la composition synchrone de deux ou plusieurs automates $(\max,+)$ (composantes) devrait être le maximum des durées dans les composantes qui participent à cette synchronisation. Dans ce but, le produit synchrone d'automates $(\max,+)$ est proposé dans la section suivante.

4. Extension des automates $(\max,+)$ par produit synchrone

Dans cette section nous allons présenter une extension de nos techniques aux classes des automates temporisés avec plusieurs horloges. Dans ce but, nous introduisons un produit synchrone d'automates $(\max,+)$ conduisant à des automates $(\max,+)$ définis sur un alphabet étendu pour la prise en compte d'occurrences simultanées d'événements.

Les automates $(\max,+)$ sont intrinsèquement un modèle séquentiel, où les événements se produisent l'un après l'autre et ne supportent pas des événements se produisant simultanément¹. La durée des mots (séquences d'événements) est en effet la somme des durées de tous les événements. Or, pour beaucoup de systèmes réels, le parallélisme (occurrence simultanée d'événements) doit être pris en compte. Dans la théorie classique des systèmes concurrents logiques (tels que les réseaux de Petri), la notion de trace a été proposée pour cela. Elle consiste à ne plus distinguer l'ordre des événements simultanés dans les séquences (mots). Les événements simultanés sont liés par une relation symétrique d'indépendance traduisant leur commutation. Il en résulte un monoïde partiellement commutatif (en anglais *trace monoid*). La théorie des traces, aussi connu sous le nom des ordres partiels, a été développée depuis une trentaine d'années (Mazurkiewicz, 1987). A notre connaissance, cette théorie n'admet pas une généralisation directe au cas temporisé.

1. Notons tout de même qu'une sous-classe des automates $(\max,+)$, appelée automates de type tas, permet de modéliser les systèmes temporisés concurrents (en l'occurrence tout réseau de Petri temporisé sauf) mais sous forme d'automates non déterministes.

Ceci nous conduit à considérer des produits synchrones d'automates (max,+) en tant qu'automates (max,+) concurrents définis sur un ensemble d'événements particuliers. Il s'agit là d'une adaptation des automates logiques multi-événements (Takai *et al.*, 2003) définis sur l'ensemble des parties des événements. Plus particulièrement, le produit synchrone de deux automates (max,+) utilise comme ensemble d'événements (alphabet) l'union des événements partagés (communs) avec toutes les couples de séquences d'événements privés. Nous allons voir qu'en fait seules les séquences d'événements privés qui ont lieu dans les automates locaux sont concrètement utilisés lors de leur produit synchrone.

Nous introduisons tout d'abord plusieurs notations utiles à la définition de ce produit synchrone.

Soient G_1 et G_2 définis sur les alphabets locaux A_1 et A_2 . Les projections naturelles associées sont notées par $P_1 : (A_1 \cup A_2)^* \rightarrow A_1^*$ et $P_2 : (A_1 \cup A_2)^* \rightarrow A_2^*$. Nous aurons aussi besoin des matrices booléennes associées aux matrices des morphismes :

$$[B\mu(a)]_{ij} = \begin{cases} e, & \text{si } [\mu(a)]_{ij} \neq \varepsilon \\ \varepsilon, & \text{sinon} \end{cases}$$

Pour alléger, nous noterons par la suite $B(a)$ plutôt que $B\mu(a)$. La notation s'étend naturellement aux mots de A^* .

4.1. Produit synchrone des automates (max,+)

Le produit concret synchrone de deux automates (max,+) utilise comme ensemble d'événements d'une part les événements partagés et d'autre part les séquences d'événements privés qui sont exécutées par les deux automates entre deux transitions de synchronisation consécutives.

La définition de ce produit met en jeu le produit de Kronecker (produit tensoriel). Rappelons que si $A = (a_{ij})$ est une matrice $m \times n$ et B est une matrice $p \times q$ sur un dioïde, alors leur *produit tensoriel* $A \otimes^t B$ est une matrice $mp \times nq$ suivante :

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

La propriété dite du produit mixte sera utile par la suite et est donc rappelée ici.

Lemme 4.1. *Pour des matrices de dimensions adéquates A, B, C, D définis sur un commutatif dioïde commutatif, on a :*

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

Définition 4.1. *Le produit synchrone des automates (max,+)*

$G_1 = (Q_1, A_1, \alpha_1, \mu_1, \beta_1)$ et $G_2 = (Q_2, A_2, \alpha_2, \mu_2, \beta_2)$, est l'automate (max,+)
défini sur l'alphabet

$$\mathcal{A} = (A_1 \cap A_2) \cup [(A_1 \setminus A_2)^* \times (A_2 \setminus A_1)]^*$$

par :

$$G_1 \parallel G_2 = \mathcal{G} = (Q_1 \times Q_2, \mathcal{A}, \alpha, \mu, \beta)$$

avec $Q_1 \times Q_2$ l'ensemble des états, \mathcal{A} l'ensemble des événements, $\alpha = \alpha_1 \otimes^t \alpha_2$
les délais initiaux, $\mu : \mathcal{A}^* \rightarrow \mathbb{R}_{max}^{|Q_1| \times |Q_2|}$ la matrice de morphisme et $\beta = \beta_1 \otimes^t \beta_2$ les
délais finaux. La matrice de morphisme est définie par :

$$\mu(v) = \begin{cases} \mu_1(v) \otimes^t B_2(v) \oplus B_1(v) \otimes^t \mu_2(v), & \text{si } v = a \in A_1 \cap A_2 \\ \mu_1(P_1(v)) \otimes^t B_2(P_2(v)) \oplus B_1(P_1(v)) \otimes^t \mu_2(P_2(v)), & \text{si } v = (P_1(v), P_2(v)) \\ & \in (A_1 \setminus A_2)^* \times (A_2 \setminus A_1)^* \end{cases}$$

L'idée sous-jacente est la suivante : les systèmes G_1 et G_2 se synchronisent sur l'ensemble des événements communs $A_1 \cap A_2$, alors qu'entre deux synchronisations successives les systèmes G_1 et G_2 exécutent des événements privés appartenant à $A \setminus (A_1 \cup A_2)$ selon les séquences respectives $P_1(v) \in (A_1 \setminus A_2)^*$ et $P_2(v) \in (A_2 \setminus A_1)^*$ (la durée dans le produit synchrone est le maximum des durées associées à ces séquences privées). Autrement dit, toutes les tâches locales de G_1 entre deux transitions synchronisantes sont en concurrence avec toutes les tâches locales de G_2 .

Plutôt que d'utiliser une relation d'indépendance (c.f. théorie des traces), nous préférons adopter cette "approche multi-événements" afin d'éviter des difficultés relatives à l'associativité rencontrées lors de l'extension temporelle de la théorie des traces. Il est possible de définir les matrices de morphisme sur les couples d'événements indépendants de telle façon que l'extension aux traces est cohérente, c'est-à-dire de sorte que matrice de morphisme soit identique pour deux mots appartenant à la même trace. Mais il y a une difficulté d'interprétation, car pour chaque mise de parenthèses on obtient, en général, une valeur différente de la matrice de morphisme correspondant aux différentes durées d'une même tâche ! Par exemple, si b et c sont des événements indépendants, alors il se peut que $\mu([bc]b) \neq \mu(b[cb])$, ce qui n'est pas acceptable, car $[bc] = [cb]$ est la même trace ! Donc, il ne suffit pas de définir de façon appropriée $\mu([bc]) = \mu([cb])$, mais, à la différence du cas logique, il nous faut aussi considérer des séquences d'événements privés de longueurs supérieures à deux. Il nous semble qu'il est impossible de pallier cette difficulté propre au cas temporel sans prendre en compte des séquences d'événements (mots finis locaux) indépendants.

Nous soulignons que pour calculer le comportement $l(w)$ d'un mot "global" $w \in (A_1 \cup A_2)^*$ seul l'ordre des événements privés propre à un automate local est important. Par exemple, seul l'ordre des événements de $A_1 \setminus A_2$ et $A_2 \setminus A_1$ est important dans un mot $w \in ((A_1 \cup A_2) \setminus (A_1 \cap A_2))^*$. C'est à dire l'ordre des événements locaux dans $P_1(w) \in (A_1 \setminus A_2)^*$ et $P_2(w) \in (A_2 \setminus A_1)^*$. Donc, toutes les séquences

dans $\{P_1^{-1}P_1(w) \cap P_2^{-1}P_2(w)\}$ ne sont pas à distinguer entre elles. Ainsi, l'inclusion de $(A_1 \setminus A_2)^* \times (A_2 \setminus A_1)^*$ dans l'alphabet "multi-événement" a pour l'effet que nous considérons seulement les couples de mots locaux correspondants $P_1(w)$ et $P_2(w)$ dans les calculs au lieu de tout $w \in ((A_1 \cup A_2) \setminus (A_1 \cap A_2))^*$. De plus, il suffit de considérer dans les produits concrets les couples de mots qui se produisent dans les systèmes G_1 et G_2 et d'inclure dans l'alphabet étendu seulement des séquences locales complètes (jusqu'aux synchronisations). Ceci est important pour assurer que l'application $\mu : \mathcal{A} \rightarrow \mathbb{R}_{\max}$ soit un morphisme. Nous allons montrer dans la sous-section suivante que le comportement du produit synchrone peut être défini pour toute séquence de \mathcal{A}^* en utilisant une application matricielle $\nu : \mathcal{A} \rightarrow \mathbb{R}_{\max}$ induite par μ .

Maintenant, chaque mot $w \in (A_1 \cup A_2)^*$ admet une décomposition $w = v_0 a_1 v_1 \dots a_n v_n$, où les $a_i \in A_1 \cap A_2$, $i = 1, \dots, n$ sont des événements partagés et les $v_i \in (A \setminus (A_1 \cap A_2))^*$, $i = 0, \dots, n$ sont des séquences représentant les mots privés. Notons que pour chaque v_i les tâches correspondantes dans les sous systèmes G_1 et G_2 sont données par $P_1(v_i)$ et $P_2(v_i)$, respectivement. Chaque mot de $(A_1 \cup A_2)^*$ peut ainsi être vu comme un élément de \mathcal{A}^* , plus précisément :

$$w = P_1(v_0) \times P_2(v_0).a_1 P_1(v_1) \times P_2(v_1) \dots a_n P_1(v_n) \times P_2(v_n)$$

est un mot sur l'alphabet étendu \mathcal{A} .

Le morphisme μ de la définition 4.1 est défini sur \mathcal{A}^* et induit également une application matricielle ν définie sur $(A_1 \cup A_2)^*$ à valeurs dans \mathbb{R}_{\max} définie par :

$$\nu(w) = \mu(v_0)\mu(a_1)\mu(v_1) \dots \mu(a_n)\mu(v_n).$$

Il est important de noter que dans les cas où l'alphabet \mathcal{A} est infini, pour chaque mot fini $w \in (A_1 \cup A_2)^*$ on a besoin de seulement un nombre fini de matrices de morphisme pour définir l'application matricielle $\nu : (A_1 \cup A_2)^* \rightarrow \mathbb{R}_{\max}$ induit par $\mu : \mathcal{A}^* \rightarrow \mathbb{R}_{\max}$.

Cette approche s'étend au cas $n > 2$. Pour $n=3$ il faut considérer quatre types de synchronisations : celles parmi tous les trois sous-systèmes et celles entre chaque couple.

On aura donc :

$$\begin{aligned} \mathcal{A} = & (A_1 \cap A_2 \cap A_3) \cup \bigcup_{i \neq j \neq k} (A_i \cap A_j) \times (A_k \setminus (A_i \cup A_j))^* \cup \\ & (A_1 \setminus (A_2 \cup A_3))^* \times (A_2 \setminus (A_1 \cup A_3))^* \times (A_3 \setminus (A_1 \cup A_2))^*. \end{aligned}$$

Les matrices de morphisme correspondantes sont :

$$\left\{ \begin{array}{l} \mu(v) = \\ \mu_1(a) \otimes^t B_2(a) \otimes^t B_3(a) \oplus B_1(a) \otimes^t \mu_2(a) \otimes^t B_3(a) \oplus B_1(a) \otimes^t B_2(a) \otimes^t \mu_3(a), \\ \text{si } v = a \in A_1 \cap A_2 \cap A_3 \\ \mu_1(v_1) \otimes^t B_2(v_2) \otimes^t B_3(v_3) \oplus B_1(v_1) \otimes^t \mu_2(v_2) \otimes^t B_3(v_3) \oplus B_1(v_1) \otimes^t B_2(v_2) \otimes^t \mu_3(v_3), \\ \text{si } v = v_1 \times v_2 \times v_3 \in (A_1 \setminus (A_2 \cup A_3))^* \times (A_2 \setminus (A_1 \cup A_3))^* \times (A_3 \setminus (A_1 \cup A_2))^* \\ \mu_1(a) \otimes^t B_2(a) \otimes^t B_3(v_3) \oplus B_1(a) \otimes^t \mu_2(a) \otimes^t B_3(v_3) \oplus B_1(a) \otimes^t B_2(a) \otimes^t \mu_3(v_3), \\ \text{si } v = a \times v_3 \in (A_1 \cap A_2) \times (A_3 \setminus (A_1 \cup A_2))^* \\ \dots \end{array} \right.$$

Finalement nous notons qu'il y a un moyen de voir les multi-événements d'une manière homogène comme triplets : e.g. $v = a \in A_1 \cap A_2 \cap A_3$ peut être vu comme $a \times a \times a$ et $v = a \times v_3$ pour un $a \in A_1 \cap A_2$ peut être vu comme $a \times a \times v_3$.

4.2. Comportement du produit synchrone

Nous nous attachons maintenant à calculer le comportement d'un produit synchrone d'automates $(\max, +)$ défini par :

$$l(G_1 \parallel G_2)(w) = \alpha \nu(w) \beta.$$

En effet, il suffit de remarquer que le produit synchrone des séries booléennes L_1 et L_2 (associées aux comportements des automates G_1 et G_2), c'est-à-dire $L_1 \parallel L_2 = P_1^{-1}(L_1) \cap P_2^{-1}(L_2)$, s'écrit tout simplement $L_1 \parallel L_2(w) = L_1(P_1 w) \otimes L_2(P_2 w)$. Notons $Z = \{\nu, B\}$ et utilisons le complément $\bar{\nu} = B$ et $\bar{B} = \nu$. Nous aurons besoin du monoïde libre Z^* . L'opération de complément s'étend par morphisme à tout mot $m \in Z^*$: pour $m = m^1 \dots m^k$, $\bar{m} = \bar{m}^1 \dots \bar{m}^k$. La convention est que pour $m \in Z$, m^1 correspond soit à ν^1 soit à B^1 (en accord avec la valeur de m) (*idem* pour les séquences de Z^*). Pour une séquence $m_i = m_i^1 \dots m_i^k \in Z^k$, $i = 1, 2$ et un mot $v = v_1 \dots v_k \in A^k$ de longueur k nous définissons $m_i(v) = m_i^1(v_1) \otimes \dots \otimes m_i^k(v_k)$ le produit "mixte" des matrices de morphismes et matrices Booléennes correspondantes. Nous rappelons que tout mot $w \in (A_1 \cup A_2)^*$ s'interprète comme un mot de longueur $2n + 1$ de A^* si w admet la décomposition suivante : $w = v_0 a_1 v_1 \dots a_n v_n$ avec $a_i \in A_1 \cap A_2$, $i = 1, \dots, n$ les événements partagés et $v_i \in ((A_1 \cup A_2) \setminus (A_1 \cap A_2))^*$, $i = 0, \dots, n$ les séquences d'événements privés.

Proposition 4.2. *Le comportement induit du produit synchrone des automates G_1 et G_2 pour un mot $w = v_0 a_1 v_1 \dots a_n v_n \in (A_1 \cup A_2)^*$ s'écrit comme suit :*

$$l(G_1 \parallel G_2)(w) = \bigoplus_{m \in Z^{2n+1}} \alpha_1 m_1(P_1(w)) \beta_1 \otimes \alpha_2 \bar{m}_2(P_2(w)) \otimes \beta_2.$$

Démonstration. Nous utilisons la formule :

$$l(G_1 \parallel G_2)(w) = \alpha \nu(w) \beta$$

et procédons par induction en utilisant l'unique décomposition du mot $w \in A^*$ en les facteurs $v_0 a_1 v_1 \dots a_n v_n$. Le cas $n = 0$ (aucune synchronisation) est assez simple. En effet, pour $w = v_0 \in ((A_1 \cup A_2) \setminus (A_1 \cap A_2))^*$ nous obtenons :

$$\begin{aligned} \alpha \nu(v_0) \beta &= \alpha \mu(v_0) \beta = \\ &= (\alpha_1 \otimes^t \alpha_2) [\mu_1(P_1(v_0)) \otimes^t B_2(P_2(v_0)) \oplus B_1(P_1(v_0)) \otimes^t \mu_2(P_2(v_0))] (\beta_1 \otimes^t \beta_2) = \\ &= (\alpha_1 \otimes^t \alpha_2) (\mu_1(P_1(v_0)) \otimes^t B_2(P_2(v_0))) (\beta_1 \otimes^t \beta_2) \oplus (\alpha_1 \otimes^t \alpha_2) (B_1(P_1(v_0)) \\ &\otimes^t \mu_2(P_2(v_0))) (\beta_1 \otimes^t \beta_2) = \alpha_1 \mu_1(P_1(v_0)) \beta_1 \otimes^t \alpha_2 B_2(P_2(v_0)) \beta_2 \oplus \\ &\alpha_1 B_1(P_1(v_0)) \beta_1 \otimes^t \alpha_2 \mu_2(P_2(v_0)) \beta_2. \end{aligned}$$

Nous avons seulement utilisé la loi de distributivité et la propriété du produit mixte (voir Propriété 4.1). Notons que le produit tensoriel dans le cas des scalaires (les deux derniers termes ci-dessus) coïncide avec le produit des nombres de \mathbb{R}_{\max} . Soient $l_1, l_2 \in \mathbb{R}_{\max}(A)$ les comportements respectifs de G_1 et G_2 . Comme $\alpha_1 \otimes \mu_1(P_1(v_0)) \otimes \beta_1 = l_1(P_1(v_0))$ et $\alpha_2 \otimes B_2(P_2(v_0)) \otimes \beta_2 = \text{suppl}_2(P_2(v_0))$ nous obtenons :

$$l_1 \| l_2 = l_1(P_1(w)) \otimes \text{suppl}(l_2)(P_2(w)) \oplus l_2(P_2(w)) \otimes \text{suppl}(l_1)(P_1(w)),$$

la forme énoncée de $l(G_1 \| G_2)(v_0)$ pour un mot de \mathcal{A}^* de longueur 1.

Supposons la formule satisfaite pour $n - 1$ dans la décomposition de w . Il suffit de montrer que pour $w = v_0 a_1 v_1 \dots a_n v_n$ nous obtenons la même formule. Ceci découle du fait que l'on multiplie $2n + 1$ termes qui ont tous la même forme, notamment la somme des deux produits tensoriels avec à gauche toutes les combinaisons de matrices de morphismes et les matrices booléennes correspondantes. Remarquons aussi que pour $v = a \in A_1 \cap A_2$ nous avons $P_1(a) = P_2(a) = a$, la matrice de morphisme admet dans les deux cas la même forme en remplaçant dans le premier cas ($v = a \in A_1 \cap A_2$) a par $P_1(a)$ ou $P_2(a)$. \square

Nous pouvons donc calculer algébriquement le comportement induit d'un produit synchrone, ce qui nous permet d'étendre nos résultats sur la commande des automates (max,+) aux produits synchrones d'automates (max,+).

La formule pour le comportement d'un produit synchrone est assez complexe. Nous soulignons que dans le cas $A_1 \cap A_2 = \emptyset$ le produit synchrone des séries $s_1 \in \mathbb{R}_{\max}(A_1)$ et $s_2 \in \mathbb{R}_{\max}(A_2)$ est toujours bien défini comme nous l'avons montré dans la preuve de la Proposition 4.2. La formule s'étend au produit de $n > 2$ séries définies sur des alphabets $A_i, i = 1, \dots, n$ tels que pour $i, j = 1, \dots, n : i \neq j$ nous avons $A_i \cap A_j = \emptyset$. Notamment,

$$\begin{aligned} l_1 \| \dots \| l_n &= l_1(P_1(w)) \otimes \text{suppl}(l_2)(P_2(w)) \otimes \dots \otimes \text{suppl}(l_n)(P_n(w)) \\ &\oplus \text{suppl}(l_1)(P_1(w)) \otimes l_2(P_2(w)) \otimes \text{suppl}(l_3)(P_3(w)) \otimes \dots \otimes \text{suppl}(l_n)(P_n(w)) \\ &\oplus \dots \oplus \text{suppl}(l_1)(P_1(w)) \otimes \dots \otimes \text{suppl}(l_{n-1})(P_{n-1}(w)) \otimes \dots \otimes l_n(P_n(w)). \end{aligned}$$

4.3. Exemple

Nous proposons ici un exemple simple qui illustre l'approche présentée dans ce papier. Considérons deux automates (max,+) G_1 et G_2 sur les alphabets $A_1 = \{a, b, d\}$ et $A_2 = \{a, c\}$, respectivement, dessinés sur la figure 1.

Leur produit synchrone est d'après la définition 4.1 l'automate (max,+) 'multi-événement' suivant :

$$\mathbf{G}_1 \| \mathbf{G}_2 = \mathcal{G} = (\mathbf{Q}_1 \times \mathbf{Q}_2, \mathcal{A}, \alpha, \mu, \beta)$$

avec $Q_1 \times Q_2$ l'ensemble des états,

$$\mathcal{A} = \{a, (b^2, c), (d, c)\} \subseteq (A_1 \cap A_2) \cup [(A_1 \setminus A_2)^* \times (A_2 \setminus A_1)^*],$$

$\alpha = \alpha_1 \otimes^t \alpha_2, \beta = \beta_1 \otimes^t \beta_2$ et finalement :

$$\nu(v) = \begin{cases} \mu_1(a) \otimes^t B_2(a) \oplus B_1(a) \otimes^t \mu_2(a), & \text{si } v = a \in A_1 \cap A_2 \\ \mu_1(bb) \otimes^t B_2(c) \oplus B_1(bb) \otimes^t \mu_2(c), & \text{si } v = (b^2, c) \\ \mu_1(d) \otimes^t B_2(c) \oplus B_1(d) \otimes^t \mu_2(c), & \text{si } v = (d, c) \end{cases}$$

$$\mu(a) = \begin{pmatrix} \varepsilon & 1 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \otimes^t \begin{pmatrix} \varepsilon & e \\ \varepsilon & \varepsilon \end{pmatrix} \oplus \begin{pmatrix} \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \otimes^t \begin{pmatrix} \varepsilon & 1 \\ \varepsilon & \varepsilon \end{pmatrix}$$

$$= \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix}$$

$$\mu((b^2, c)) = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 1 \\ 3 & \varepsilon & \varepsilon \end{pmatrix} \otimes \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 1 \\ 3 & \varepsilon & \varepsilon \end{pmatrix} \otimes^t \begin{pmatrix} \varepsilon & \varepsilon \\ e & \varepsilon \end{pmatrix}$$

$$\oplus \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e \\ e & \varepsilon & \varepsilon \end{pmatrix} \otimes \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e \\ e & \varepsilon & \varepsilon \end{pmatrix} \otimes^t \begin{pmatrix} \varepsilon & \varepsilon \\ 5 & \varepsilon \end{pmatrix}$$

$$= \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \oplus \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix}$$

$$= \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix}$$

et

$$\begin{aligned}
 \mu((d,c)) &= \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ 10 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \otimes^t \begin{pmatrix} \varepsilon & \varepsilon \\ e & \varepsilon \end{pmatrix} \oplus \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \otimes^t \begin{pmatrix} \varepsilon & \varepsilon \\ 5 & \varepsilon \end{pmatrix} \\
 &= \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 10 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \oplus \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \\
 &= \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 10 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix}
 \end{aligned}$$

En accord avec la proposition 4.2 nous obtenons pour le mot $w = abc$ le comportement induit $l(abc) = \alpha\mu(a)\mu(bbc)\beta = 6$. Similairement, pour $w = a(bbc)a(cd)$ nous obtenons : $l(abbcaacd) = \alpha\mu(a)\mu(bbc)\mu(a)\mu(dc)\beta = 17$.

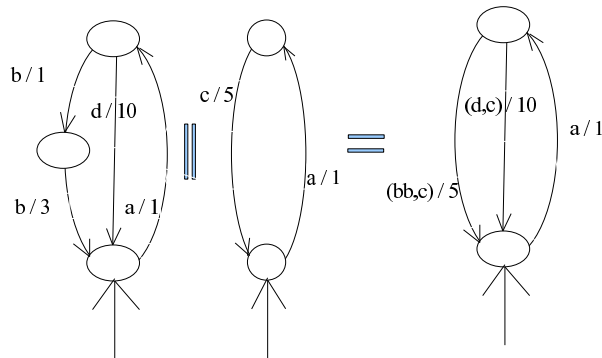


Figure 1. Les automates G_1 , G_2 et $G_1||G_2$

4.4. Interprétation en termes des réseaux de Petri temporisés

Finalement, nous allons expliquer notre démarche en termes des réseaux de Petri temporisés. Considérons le même exemple qu'à la section précédente.

Les réseaux de Petri temporisés correspondants sont sur la figure 2.

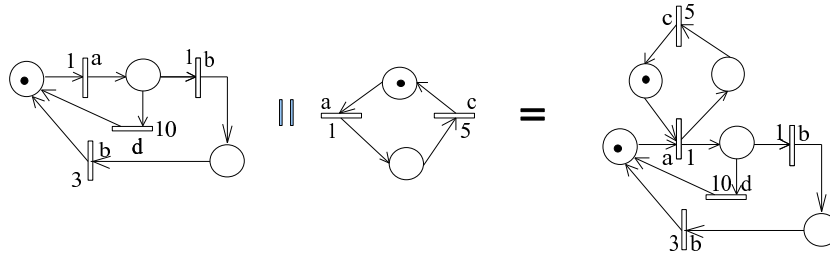


Figure 2. Les réseaux de Petri N_1 , N_2 et $N_1||N_2$.

L'idée de base derrière notre approche est donc de composer des graphes d'états temporisés afin de modéliser des synchronisations dans les réseaux de Petri à l'aide des automates $(\max,+)$ déterministes.

Nous remarquons encore que dans le cas général pour le calcul du produit synchrone de deux (ou plusieurs) séries, ces séries elles mêmes ne suffisent pas. Leurs représentations linéaires sont donc nécessaires. Ceci est nécessaire dès qu'il y a un comportement à calculer dans des mots de longueurs supérieures ou égales à deux sur l'alphabet étendu \mathcal{A} .

Ceci n'est pas gênant dans notre approche, car nous travaillons typiquement avec des séries déterministes (comportements de systèmes déterministes comme des graphes d'états temporisés avec leurs transitions labelées injectivement et donc leurs graphes de marquage temporisés qui sont des automates $(\max,+)$ déterministes). La théorie de réalisation des séries déterministes est satisfaisante (on peut obtenir des représentations linéaires sous formes d'automates $(\max,+)$). Il n'y a alors plus de problèmes avec la décidabilité des inégalités quand on travaille avec des séries déterministes ((Lombardy *et al.*, 2006)). Mais dans ce cas il faut bien utiliser le produit synchrone afin de pouvoir modéliser le phénomène de synchronisation (sinon \oplus n'est pas utilisé dans les automates $(\max,+)$ déterministes).

Remerciements

Ce travail a été soutenu par l'Académie tchèque des Sciences, projet de recherche No. AV0Z10190503, par l'université d'Angers et par le projet européen EU.ICT N.224498 (DISC).

5. Conclusion

Nous avons défini le produit synchrone des automates $(\max,+)$ et proposé l'expression du comportement induit.

Ce papier ouvre en particulier la voie pour développer la commande décentralisée des automates (max,+). Cette commande permettrait de réduire la complexité de la commande de systèmes répartis temporisés de grande taille. Nous envisageons également une extension qui prendra en compte les durées nondéterministes de transitions, comprises entre une borne inférieure et une borne supérieure à l'aide d'un diode des intervalles.

Dans la suite de nos travaux nous pourrions prendre en compte les événements incontrôlables en considérant des événements localement incontrôlables et étudier la commande décentralisée des automates (max,+), à commencer par le cas relativement simple avec des alphabets disjoints (les sous-systèmes étant liés implicitement à travers une spécification définie sur l'alphabet global).

6. Bibliographie

- Arnold A., *Finite Transition Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1994.
- Baccelli F., Cohen G., Olsder G., Quadrat J.-P., *Synchronization and Linearity. An Algebra for Discrete Event Systems*, New York, Wiley, 1992.
- Gaubert S., « Performance Evaluation of (max,+) Automata », *IEEE Transaction on Automatic Control*, vol. 40, n° 12, p. 2014-2025, 1995.
- Gaubert S., Mairesse J., « Modeling and Analysis of Timed Petri Nets using Heaps of Pieces », *IEEE Transaction on Automatic Control*, vol. 44, n° 4, p. 683-698, 1999.
- Komenda J., Lahaye S., Boimond J.-L., « Supervisory Control of (max,+) Automata : A single Step Approach », *Proceedings of European Control Conference (ECC) 2009*, EUCA, Budapest, 23-26 Août, 2009.
- Lombardy S., Mairesse J., « Series which are both Max-plus and Min-plus Rational are Unambiguous », *RAIRO - Theoretical Informatics and Applications*, vol. 40, n° 1, p. 1-14, 2006.
- Mazurkiewicz A., « Trace theory », *LNCS : Petri Nets, Applications and Relationship to other Models of Concurrency*, vol. 255, n° 1, p. 279-324, 1987.
- Ramadge P., Wonham W., « The Control of Discrete-Event Systems », *Proc. IEEE*, vol. 77, n° 1, p. 81-98, 1989.
- Takai S., Ushio T., « Supervisor synthesis for a class of concurrent discrete event systems », *Proceedings of the 42nd IEEE Conference on Decision and Control*, IEEE, p. 2686-2691, 9-12 Décembre, 2003.
- Wong-Toi H., Hoffmann G., *The Control of Dense Real-Time Discrete Event Systems*, Technical Report n° STA-CS-92-1411, Stanford University, 1995.