

Modélisation modulaire à l'aide d'automates $(\max, +)$ ^{*}

Sébastien Lahaye ^{*} Jan Komenda ^{**} Jean-Louis Boimond ^{*}

^{*} *LUNAM Université, LISA, Angers, France ([sebastien.lahaye,
jean-louis.boimond]@univ-angers.fr).*

^{**} *Institute of Mathematics - Brno Branch, Czech Academy of
Sciences, Czech Republic (komenda@ipm.cz).*

Résumé Une démarche modulaire pour la modélisation de systèmes à événements discrets à l'aide d'automates $(\max, +)$ est proposée. Elle consiste à décomposer le système en sous-systèmes, chacun pouvant être modélisé par un automate $(\max, +)$ déterministe. Seuls des phénomènes de concurrence sont ainsi modélisés. Les interactions entre ces sous-systèmes sont prises en compte au travers d'événements communs survenant simultanément au sein de chaque composante, et sont traduites par un produit synchrone entre les différents automates déterministes. Des phénomènes de synchronisation peuvent ainsi être modélisés. On met en avant que cette démarche revient, en termes de réseaux de Petri, à modéliser des sous-systèmes à l'aide de graphes d'état temporisés saufs et à composer ceux-ci en fusionnant des transitions.

Keywords: Système à événements discrets, modélisation, automate $(\max, +)$, produit synchrone.

1. INTRODUCTION

Les automates à multiplicités dans le semi-anneau $(\max, +)$ constituent une classe d'automates temporisés. Dans Gaubert and Mairesse (1999), les auteurs ont étudié leur puissance de modélisation pour les systèmes à événements discrets : le fonctionnement de tout réseau de Petri temporisé sauf ¹ peut être retranscrit par un automate de type tas, à savoir un automate $(\max, +)$ particulier ². Ce formalisme, qui s'adresse à une classe large de systèmes à événements discrets, a été appliqué avantageusement pour l'évaluation de performances Gaubert (1995); Su and Woeginger (2011), l'ordonnancement Houssin (2011) et la commande Komenda et al. (2009b). La contribution présente se situe uniquement au niveau de la modélisation et poursuit deux objectifs principaux :

- rendre plus abordable la modélisation de systèmes à événements discrets à l'aide d'automates $(\max, +)$;
- permettre d'envisager subséquemment une commande modulaire pour les automates $(\max, +)$.

En effet, on propose ici une démarche modulaire consistant à modéliser le système en différentes composantes, chacune étant décrite par un automate $(\max, +)$ déterministe, interagissant entre elles au travers d'événements communs. On considère que ces événements communs à plusieurs composantes surviennent simultanément dans celles-ci, et l'interaction est donc définie au travers d'un produit qua-

lifié de synchrone. Pour cerner au mieux l'étendue de cette étude, on s'attache à préciser les analogies en termes de réseaux de Petri. Plus précisément, on montre tout d'abord qu'un automate $(\max, +)$ déterministe a la même puissance de modélisation qu'un graphe d'état temporisé sauf. On montre ensuite que le produit synchrone proposé pour les automates $(\max, +)$ déterministes est l'analogue de la fusion de transitions entre les graphes d'état temporisés saufs correspondants.

Le résultat du produit synchrone est un automate $(\max, +)$ non déterministe défini sur l'union des alphabets. A notre connaissance, cette approche pour la composition synchrone d'automates est originale. Il faut souligner que le nombre d'états dans l'automate résultant est égal à la somme du nombre d'états dans les automates initiaux. Ceci démarque singulièrement l'approche de celle bien admise pour les automates à états finis (pour lesquels la fonction de transition prend des valeurs booléennes, voir par exemple Cassandras and Lafortune (2006)) voire d'autres approches pour le produit synchrone d'automates $(\max, +)$ Buchholz and Kemper (2003); Su et al. (2012), pour lesquelles le nombre d'états dans l'automate composé est plutôt égal au produit du nombre d'états dans les automates initiaux.

Notons qu'une composition synchrone d'automates $(\max, +)$ a été proposée dans Komenda et al. (2009a) sous la forme d'un automate défini sur un alphabet étendu qui est construit à partir de ceux des différentes composantes. Cet alphabet étendu regroupe les événements communs ainsi que l'ensemble des séquences d'événements privées qui peuvent survenir dans l'intervalle d'événements communs. L'automate produit défini sur cet alphabet est déterministe. A notre connaissance, il n'y a pas de raison de privilégier cette opération vis-à-vis du produit syn-

^{*}. Ce travail a été partiellement soutenu par le projet de recherche AV0Z10190503 et par le projet EU.ICT N.224498 (DISC).

1. Ou encore 1-borné, c'est-à-dire que toute place contient au plus un jeton.

2. Notons que le caractère sauf est une condition suffisante pour que le fonctionnement du réseau de Petri puisse être décrit par un automate $(\max, +)$, et qu'il est donc possible qu'une classe encore plus large puisse être appréhendée.

chrone proposé ici, et il est probable que ces opérations soient complémentaires (que chacune trouve un cadre d'utilisation spécifique).

Le papier est organisé comme suit. La section suivante est dédiée à des rappels et précisions nécessaires à notre approche. Dans la Section 3, on précise le pouvoir de modélisation d'un automate $(\max, +)$ déterministe. La Section 4 est dédiée à la définition du produit synchrone d'automates déterministes. On y précise également la capacité de modélisation obtenue par ce biais.

2. PRÉLIMINAIRES

Dans cette section, on fait des rappels succincts sur les semi-anneaux idempotents, les automates $(\max, +)$ et les réseaux de Petri. L'objectif est uniquement de préciser les définitions et notations utiles pour la suite. Le lecteur est invité à consulter respectivement les références Baccelli et al. (1992), Gaubert (1995) et David and Alla (1989) pour des présentations complètes.

Un *diïde* est un *semi-anneau* dans lequel la somme, notée \oplus , est idempotente. La somme (resp. le produit, noté \otimes) admet un élément neutre noté ε (resp. e).

Exemple 1. L'ensemble $(\mathbb{R} \cup \{-\infty\})$ muni de l'opérateur \max comme somme et de l'addition usuelle comme produit est un diïde noté \mathbb{R}_{\max} avec $e = 0$ et $\varepsilon = -\infty$. L'ensemble des matrices de dimension $n \times n$ à coefficients dans \mathbb{R}_{\max} , équipé de la somme et du produit matriciels définis de façon conventionnelle à partir de \oplus et \otimes , est aussi un diïde noté $\mathbb{R}_{\max}^{n \times n}$. L'élément neutre pour l'addition de ce diïde matriciel est la matrice composée exclusivement de $\varepsilon = -\infty$. On note I_n l'élément neutre du produit, à savoir la matrice composée de $e = 0$ sur la diagonale et de $\varepsilon = -\infty$ partout ailleurs. Les vecteurs de dimension $1 \times n$ peuvent être plongés dans ce diïde en leur adjoignant $n - 1$ lignes remplies de ε (dans la suite, il nous arrivera, par abus et pour alléger la présentation, de faire l'économie de cette construction).

Exemple 2. 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 diïde, noté $(Pwr(A^*), \cup, \cdot)$. Le langage nul est noté $0 = \{\}$, le langage unité est noté $1 = \{\epsilon\}$ où ϵ est le mot vide.

Les automates à multiplicités dans le semi-anneau \mathbb{R}_{\max} sont appelés automates $(\max, +)$.

Définition 1. Un automate $(\max, +)$ est un quadruplet $G = (Q, A, \alpha, \mu)$ où ³

- Q et A sont les ensembles finis d'états et d'événements;
- $\alpha \in \mathbb{R}_{\max}^{1 \times |Q|}$ est tel que $\alpha_q \neq \varepsilon$ si q est un état initial;
- $\mu : A^* \rightarrow \mathbb{R}_{\max}^{|Q| \times |Q|}$ est un morphisme spécifié de façon unique par la famille de matrices $\mu(a) \in \mathbb{R}_{\max}^{|Q| \times |Q|}$, $a \in A$, et pour un mot $w = a_1 \dots a_n$, on a

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

3. Pour alléger la présentation et sans perte de généralité, la définition adoptée omet de distinguer les états marqués de l'automate.

où le produit matriciel mis en jeu est celui de $\mathbb{R}_{\max}^{|Q| \times |Q|}$. Un coefficient $[\mu(a)]_{q'q} \neq \varepsilon$ signifie que, dans l'état q' , l'occurrence de l'événement a provoque une transition d'état vers q .

Dans la suite, on se restreint à des automates pour lesquels les "délais initiaux", c'est-à-dire les coefficients de α différents de ε , sont tous égaux à $e = 0$.

Exemple 3. La figure 1 est un exemple de représentation graphique ⁴ qui peut être associée à chaque automate $(\max, +)$:

- les sommets correspondent aux états $q \in Q$;
- un arc existe d'un état $q' \in Q$ à un état q s'il existe $a \in A$ tel que $[\mu(a)]_{q'q} \neq \varepsilon$: celui-ci représente la transition d'état lorsque l'événement a survient et la valeur de $[\mu(a)]_{q'q}$ s'interprète comme la temporisation associée alors à a (c'est-à-dire le laps de temps durant lequel a doit être validé avant qu'il puisse survenir);
- un arc d'entrée signifie un état initial.

Pour cet exemple, on a $Q = \{I, II\}$, $A = \{a, b, c\}$, et

$$\alpha = (e \ \varepsilon), \mu(a) = \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & \varepsilon \end{pmatrix},$$

$$\mu(b) = \begin{pmatrix} \varepsilon & \varepsilon \\ 1 & \varepsilon \end{pmatrix}, \mu(c) = \begin{pmatrix} \varepsilon & \varepsilon \\ \varepsilon & 3 \end{pmatrix}.$$

Les séquences d'événements possibles sont les mots : $a, ac, ab, acc, acb, aba, accc, accb, acba, abac, abab, \dots$

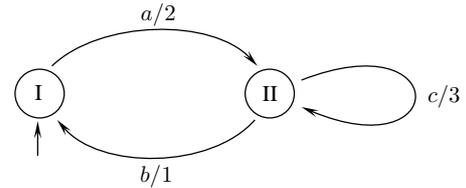


FIGURE 1. Un automate $(\max, +)$.

Un automate $(\max, +)$ est dit *déterministe* si

- il comporte un unique état initial, c'est-à-dire qu'il existe un unique $q \in Q$ tel que $\alpha_q \neq \varepsilon$;
- depuis chaque état, l'occurrence d'un événement ne peut pas induire plusieurs transitions d'état possibles, c'est-à-dire, si pour tout $a \in A$ chaque ligne de $\mu(a)$ contient au plus un élément différent de ε .

On définit $x_G(w) \in \mathbb{R}_{\max}^{1 \times |Q|}$ par

$$x_G(w) = \alpha \mu(w).$$

Un élément $[x_G(w)]_q$ s'interprète comme la date à laquelle l'état q est atteint consécutivement à la séquence d'événements w (avec la convention $[x_G(w)]_q = \varepsilon$ si l'état q n'est pas atteint à l'issue de w). On a

$$\begin{cases} x_G(\epsilon) = \alpha \\ x_G(wa) = x_G(w)\mu(a) \end{cases} \quad (1)$$

Définition 2. (Réseau de Petri). Un *réseau de Petri* est un quadruplet $\mathcal{G} = (\mathcal{P}, \mathcal{T}, \mathcal{F}, M)$, pour lequel \mathcal{P} est un ensemble fini de places (représentées par des cercles), \mathcal{T} est un ensemble fini de transitions (représentées par des

4. Cette représentation graphique met en évidence que les automates $(\max, +)$ peuvent être vus comme des automates logiques (tels qu'ils ont été définis dans Ramadge and Wonham (1989)) dans lesquels le temps est intégré, c'est-à-dire, comme une classe d'automates temporisés.

rectangles), $\mathcal{F} \subseteq (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$ est une relation entre les places et les transitions (représentée par des arcs orientés), $M : \mathcal{P} \rightarrow \mathbb{N}$ définit le *marquage initial* des places (représenté par des jetons dans les places).

Le marquage évolue selon la règle suivante :

- (1) Une transition $a \in \mathcal{T}$ est *validée* lors du marquage M si au moins un jeton est contenu dans chacune des places d'entrée.
- (2) Une transition validée a peut être franchie. Le franchissement de a transforme le marquage M en un marquage M' en retirant un jeton de chacune des places d'entrée et en ajoutant un jeton dans chacune des places de sortie de a .

Un réseau de Petri est dit *sauf* si pour tout marquage accessible à partir de M , chaque place contient au plus un jeton. Pour une transition $a \in \mathcal{T}$, on note $\bullet a$ (respectivement a^\bullet) l'ensemble des places d'entrée (respectivement, de sortie) pour a . Si pour toutes les transitions, ces ensembles sont des singletons, alors le réseau de Petri est un *graphe d'état*. Dans un graphe d'état sauf, pour tout marquage accessible, il y a une seule place qui contient un jeton.

On considère dans ce qui suit des réseaux de Petri temporisés au niveau des transitions. Une durée finie de franchissement τ est alors associée avec chaque transition : τ est le temps minimal durant lequel une transition doit être validée avant de pouvoir être franchie.

Plusieurs hypothèses sont considérées dans la suite concernant le fonctionnement des réseaux de Petri :

- un jeton du marquage initial est considéré être introduit dans le réseau de Petri à la date 0 ;
- une transition est franchie dès que possible, c'est-à-dire dès qu'elle a été validée pendant le temps de sa temporisation (hypothèse de *fonctionnement au plus tôt*) ;
- si une place possède plusieurs transitions de sortie, alors pour chaque jeton il y a un *conflit* entre les transitions qui peuvent concurremment être franchies. Dans ce travail, lors d'un conflit, tous les choix logiques possibles sont considérés (à l'inverse de la *politique de la course* souvent considérée dans la littérature, avec laquelle une transition est choisie sur des considérations temporelles).

On définit $x_{\mathcal{G}}(w) \in \mathbb{R}_{\max}^{1 \times |\mathcal{P}|}$, un vecteur de variables associées aux places $q \in \mathcal{P}$ et dépendant de la séquence de franchissements $w \in \mathcal{T}^*$ ayant eu lieu, par

$$[x_{\mathcal{G}}(w)]_q = \quad (2)$$

$$\begin{cases} \text{date d'arrivée du dernier jeton encore contenu dans } q, \\ \varepsilon \text{ si } q \text{ ne contient pas de jeton.} \end{cases}$$

3. MODÉLISATION À L'AIDE D'AUTOMATES (MAX, +) DÉTERMINISTES

Il est pris ici comme point de vue que la "brique élémentaire" de modélisation⁵ est l'automate (max, +) déterministe. Dans la proposition suivante, on cerne le "périmètre" de modélisation ainsi considéré.

5. Dans le sens où le modèle d'un système va être obtenu à partir d'un assemblage de telles briques élémentaires.

Proposition 1. Automates (max, +) déterministes et graphes d'état temporisés saufs constituent des classes de systèmes à événements discrets équivalentes.

Proof. On commence par montrer que tout graphe d'état temporisé sauf $\mathcal{G} = (\mathcal{P}, \mathcal{T}, \mathcal{F}, M)$ peut être transformé en un automate (max, +) déterministe $G = (Q, A, \alpha, \mu)$ avec

$$x_{\mathcal{G}}(w) = x_G(w),$$

pour tout w correspondant respectivement à une séquence de franchissements dans \mathcal{G} et une séquence de transitions d'états dans G . La transformation consiste à :

- (1) faire correspondre un état dans Q à chaque place dans \mathcal{P} ;
- (2) faire correspondre l'état initial (i.e., l'unique $q \in Q$ tel que $\alpha_q \neq \varepsilon$ et $\alpha_q = e$) à la seule place contenant un jeton (i.e., l'unique $q \in \mathcal{P}$ tel que $M(q) \neq 0$ et $M(q) = 1$) ;
- (3) associer un événement $a \in A$ à chaque transition $a \in \mathcal{T}$. Le morphisme de G est alors défini par $\forall a \in A, \forall q, q' \in Q,$

$$[\mu(a)]_{qq'} = \begin{cases} \tau & \text{si } q = \bullet a, q' = a^\bullet \text{ et la tempo. de } a \text{ vaut } \tau, \\ \varepsilon & \text{sinon.} \end{cases}$$

Puisque \mathcal{G} est un graphe d'état, $\bullet a$ et a^\bullet sont des singletons, et le morphisme ainsi défini est tel que pour chaque q , il existe au plus un q' tel que $[\mu(a)]_{qq'} \neq \varepsilon$. En d'autres termes, l'automate ainsi dérivé est déterministe.

Notons tout d'abord que pour le mot vide $w = \varepsilon$, (i.e. en conditions initiales), on a par construction

$$[x_{\mathcal{G}}(\varepsilon)]_q = \alpha_q = [x_G(\varepsilon)]_q = \begin{cases} 0 & \text{si } M(q) = 1, \\ \varepsilon & \text{sinon.} \end{cases}$$

Supposons que l'égalité

$$x_{\mathcal{G}}(w) = x_G(w),$$

soit vraie à l'issue d'une séquence de franchissements (respectivement, une séquence de transitions d'états) w dans \mathcal{G} (respectivement, dans G), et vérifions que $x_{\mathcal{G}}(wa) = x_G(wa), \forall a$. Puisque \mathcal{G} est un graphe d'état, $\bullet a$ et a^\bullet sont des singletons et on note $q = \bullet a$ et $q' = a^\bullet$. Soit τ la temporisation associée à a , celle-ci est validée à partir de l'instant $[x_{\mathcal{G}}(w)]_q$ et franchie à la date $\tau + [x_{\mathcal{G}}(w)]_q$, d'où

$$[x_{\mathcal{G}}(wa)]_{q'} = \tau + [x_{\mathcal{G}}(w)]_q.$$

Seule q' contient un jeton à l'issue de la séquence de franchissements wa et donc $[x_{\mathcal{G}}(wa)]_i = \varepsilon$ pour $i \in \mathcal{P}$ et $i \neq q'$. Dans l'automate G obtenu comme ci-dessus à partir de \mathcal{G} , on a par construction $[\mu(a)]_{qq'} = \tau$ et c'est le seul coefficient de $\mu(a)$ différent de ε . D'après (1), on a $x_G(wa) = x_G(w)\mu(a)$, d'où $[x_G(wa)]_{q'} = \tau + [x_G(w)]_q$ et $[x_G(wa)]_i = \varepsilon$ pour $i \in Q$ et $i \neq q'$.

De façon symétrique, on peut transformer un automate (max, +) déterministe $G = (Q, A, \alpha, \mu)$ en un graphe d'état temporisé sauf $\mathcal{G} = (\mathcal{P}, \mathcal{T}, \mathcal{F}, M)$. Seule l'étape (3) ci-dessus n'est pas exactement symétrique dans la construction, à savoir qu'à chaque triplet $(q, q', a), q, q' \in Q, a \in A$ tel que $[\mu(a)]_{qq'} \neq \varepsilon$ est associée une transition ($\in \mathcal{T}$) dans \mathcal{G} avec $[\mu(a)]_{qq'}$ comme temporisation. Par une argumentation similaire à celle ci-dessus, on peut vérifier que

$$x_G(w) = x_{\mathcal{G}}(w),$$

pour tout w .

Exemple 4. Sur la figure 2, on a représenté le graphe d'état temporisé sauf équivalent à l'automate $(\max, +)$ déterministe de la figure 1.

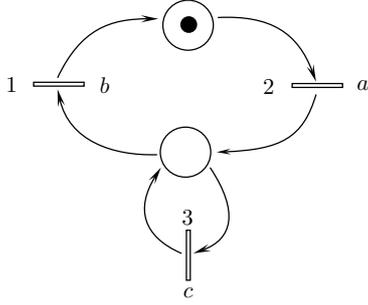


FIGURE 2. Un graphe d'état temporisé sauf.

4. COMPOSITION D'AUTOMATES $(\max, +)$ DÉTERMINISTES

On définit maintenant la composition synchrone de deux automates $(\max, +)$ déterministes (correspondant aux briques de modélisation) $G_1 = (Q_1, A_1, \alpha_1, \mu_1)$ et $G_2 = (Q_2, A_2, \alpha_2, \mu_2)$. Pour simplifier, on considère qu'à un événement commun $a \in A_1 \cap A_2$, une même valeur de temporisation est associée au sein de chaque automate.

Définition 3. On note $G_1 || G_2 = (Q, A, \alpha, \mu)$ l'automate résultant du produit synchrone de G_1 et G_2 défini par :

$$Q = Q_1 \cup Q_2, A = A_1 \cup A_2, \alpha = (\alpha_1 \ \alpha_2),$$

$$\mu(a) = \begin{pmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{pmatrix},$$

où chaque bloc μ_{ij} est une matrice de dimension $|Q_i| \times |Q_j|$ définie comme suit :

– pour $i = j$:

$$\mu_{ii} = \begin{cases} \mu_i(a), & \text{si } a \in A_i; \\ I_{|Q_i|}, & \text{sinon.} \end{cases}$$

– pour $i \neq j$:

$$[\mu_{ij}]_{kl} = \begin{cases} \tau, & \text{si } a \in A_1 \cap A_2 \text{ et } \exists q'_k \in Q_i, q'_l \in Q_j \\ & \text{t.q. } [\mu_i(a)]_{q_k, q'_k} = [\mu_j(a)]_{q'_l, q_l} = \tau (\neq \varepsilon); \\ \varepsilon, & \text{sinon.} \end{cases}$$

Exemple 5. Notons G_1 et G_2 les automates $(\max, +)$ déterministes représentés respectivement sur les figures 1 et 3. Le produit synchrone est définie sur l'alphabet

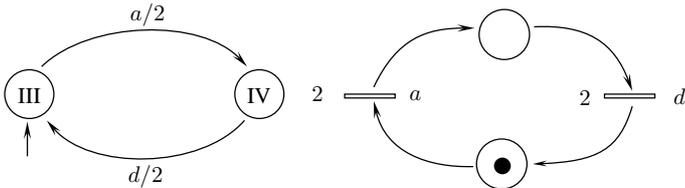


FIGURE 3. Automate G_2 et graphe d'état G_2 .

$A = A_1 \cup A_2 = \{a, b, c, d\}$. Par exemple, pour l'événement

privé $b \in A_1 \setminus (A_1 \cap A_2)$ on a (pour alléger les notations, on remplace dans les matrices suivantes les ' ε ' par des ' \cdot ') :

$$\mu(b) = \begin{pmatrix} \mu_1(b) & \cdot \\ \cdot & I_{|Q_2|} \end{pmatrix} = \begin{pmatrix} \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot \\ \cdot & e & \cdot \\ \cdot & \cdot & e \end{pmatrix},$$

Pour interpréter la construction de ces matrices, rappelons que les lignes et colonnes sont indexés sur l'ensemble des états, ici $\{I, II, III, IV\}$. Dans la composition synchrone, l'occurrence de b

- induit une transition de l'état II vers l'état I au sein de G_1 après une temporisation de 1 unité de temps, ce qui est traduit par le terme $[\mu(b)]_{II, I} = 1$;
 - n'a aucune influence au sein de G_2 (car il est privé à G_1). Pour traduire cela, les termes $[\mu(b)]_{III, III} = e$ et $[\mu(b)]_{IV, IV} = e$ "codent" une transition depuis chaque état de G_2 vers lui-même avec une temporisation nulle.
- Pour l'événement commun a , on a

$$\mu(a) = \begin{pmatrix} \cdot & 2 & \cdot & 2 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & 2 & \cdot & 2 \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}.$$

La construction de la matrice $\mu(a)$ s'interprète comme suit :

- le terme $[\mu(a)]_{I, II} = 2$ (resp., $[\mu(a)]_{III, IV} = 2$) "code" la transition de l'état I vers l'état II (resp., de l'état III vers l'état IV) au sein de G_1 (resp., G_2) après une temporisation de 2 unités de temps;
- pour traduire la simultanéité de l'occurrence de a au sein de G_1 et G_2
 - le terme $[\mu(a)]_{I, IV} = 2$ transcrit le fait que la transition depuis l'état I dans G_1 conduit dans le même temps à l'état IV au sein de G_2 après une temporisation de 2 unités de temps;
 - le terme $[\mu(a)]_{III, II} = 2$ transcrit de façon symétrique la simultanéité de la transition depuis l'état III dans G_1 et de la transition vers l'état IV dans G_2 après 2 unités de temps.

On a d'autre part

$$\alpha = (\alpha_1 | \alpha_2) = (e \ \cdot | e \ \cdot).$$

La figure 4 est une représentation de l'automate $G_1 || G_2$. Il est non déterministe : depuis certains états plusieurs transitions sont possibles sur l'occurrence d'un même événement et il y a deux états initiaux.

Remarque 1.

- (1) La définition 3 s'étend sans difficulté pour pouvoir considérer la composition synchrone de plus de deux automates $(\max, +)$.
- (2) Le nombre d'états dans l'automate résultant du produit est égal à la somme du nombre d'états dans les automates initiaux. Dans la littérature sur les automates Cassandras and Lafortune (2006) et les automates $(\max, +)$ Buchholz and Kemper (2003); Su et al. (2012), les compositions sont plutôt telles que le nombre d'états dans l'automate composé est égal au produit du nombre d'états dans les automates initiaux.
- (3) On peut se demander s'il existe un automate $(\max, +)$ déterministe équivalent à $G_1 || G_2$. Il est bien connu

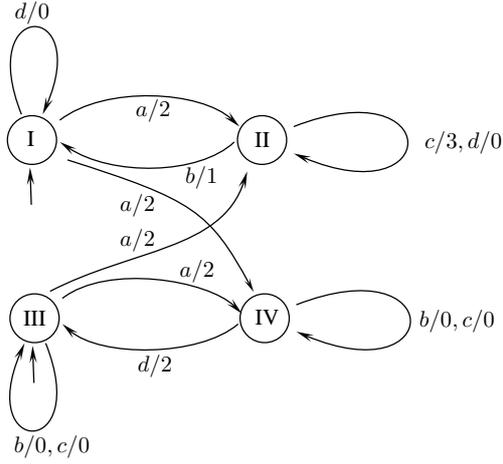


FIGURE 4. Automate (max,+) $G_1||G_2$

(voir par exemple Cassandras and Lafortune (2006)) que tout automate non déterministe admet un automate déterministe équivalent (ayant le même langage). Pour les automates (max, +), la réponse à la *déterminisation* est moins catégorique et a fait l'objet de plusieurs travaux (sans être exhaustif, citons Gaubert (1995); Klimann et al. (2004); Lombardy and Sakarovich (2006); Kirsten and Lombardy (2009)). Pour le produit synchrone $G_1||G_2$, ces travaux n'apportent pas, à notre connaissance, une réponse immédiate et cela constitue donc une piste à explorer.

La proposition suivante souligne les capacités de modélisation obtenues en faisant le produit synchrone d'automates (max, +) déterministes. Celles-ci s'expriment au travers d'un réseau de Petri résultant de la *fusion* de transitions communes entre deux sous-réseaux : par exemple, le réseau de Petri sur la figure 5 est le résultat de la fusion de la transition a commune aux réseaux des figures 2 et 3.

Proposition 2. Soient \mathcal{G}_1 et \mathcal{G}_2 les graphes d'état temporisés saufs équivalents aux automates (max, +) déterministes G_1 et G_2 . On note $\mathcal{G}_1||\mathcal{G}_2$ le réseau de Petri obtenu à partir de \mathcal{G}_1 et \mathcal{G}_2 en fusionnant les transitions correspondant aux événements communs $a \in A_1 \cap A_2$ de G_1 et G_2 . On a :

$$x_{\mathcal{G}_1||\mathcal{G}_2}(w) = x_{G_1||G_2}(w),$$

pout tout $w \in A^*$ traduisant une séquence de franchissements dans $\mathcal{G}_1||\mathcal{G}_2$ telle que $P_1(w)$ (P_1 désigne la projection sur le langage défini à partir de A_1 ⁶) et $P_2(w)$ soient des séquences de franchissements possibles dans \mathcal{G}_1 et \mathcal{G}_2 (en d'autres termes, $P_1(w)$ et $P_2(w)$ doivent appartenir aux langages de \mathcal{G}_1 et \mathcal{G}_2).

Proof. On a par construction $x_{\mathcal{G}_1||\mathcal{G}_2}(\epsilon) = x_{G_1||G_2}(\epsilon)$. Supposons que l'égalité $x_{\mathcal{G}_1||\mathcal{G}_2}(w) = x_{G_1||G_2}(w)$ soit vraie à l'issue d'une séquence de franchissements (resp., une séquence de transitions d'états) w dans $\mathcal{G}_1||\mathcal{G}_2$ (resp., dans $G_1||G_2$), et vérifions que $x_{\mathcal{G}_1||\mathcal{G}_2}(wa) = x_{G_1||G_2}(wa)$, $\forall a$.

6. La projection naturelle, de A vers $A_1 \subseteq A$, est telle que :

$$P_1(a) = \begin{cases} a & \text{si } a \in A_1, \\ \epsilon & \text{si } a \in A \setminus A_1, \end{cases}$$

P_1 préserve la concaténation et s'étend aux mots (pour chaque mot, P_1 ôte les événements (lettres) de $A \setminus A_1$).

Considérons une transition correspondant à un événement commun $a \in A_1 \cap A_2$, c'est-à-dire, une transition $a \in \mathcal{T}$ dans $\mathcal{G}_1||\mathcal{G}_2$ résultant de la fusion de transitions de \mathcal{G}_1 et \mathcal{G}_2 . Elle peut être franchie seulement si la place d'entrée dans \mathcal{G}_1 , notée q_1 ($q_1 \in \bullet a$ et $q_1 \in \mathcal{P}_1$), et la place d'entrée dans \mathcal{G}_2 , notée q_2 ($q_2 \in \bullet a$ et $q_2 \in \mathcal{P}_2$), contiennent un jeton à l'issue de la séquence w . Ces jetons contribuent à la validation de a à partir des instants respectifs $[x_{\mathcal{G}_1||\mathcal{G}_2}(w)]_{q_1}$ et $[x_{\mathcal{G}_1||\mathcal{G}_2}(w)]_{q_2}$. Ainsi on a :

$$[x_{\mathcal{G}_1||\mathcal{G}_2}(wa)]_{q'_1} = \tau + \max([x_{\mathcal{G}_1||\mathcal{G}_2}(w)]_{q_1}, [x_{\mathcal{G}_1||\mathcal{G}_2}(w)]_{q_2}), \quad (3)$$

$$[x_{\mathcal{G}_1||\mathcal{G}_2}(wa)]_{q'_2} = \tau + \max([x_{\mathcal{G}_1||\mathcal{G}_2}(w)]_{q_1}, [x_{\mathcal{G}_1||\mathcal{G}_2}(w)]_{q_2}), \quad (4)$$

pour $a^\bullet = \{q'_1, q'_2\}$ avec $q'_1 \in \mathcal{P}_1$, $q'_2 \in \mathcal{P}_2$, et

$$[x_{\mathcal{G}_1||\mathcal{G}_2}(wa)]_{q'} = \epsilon \quad (5)$$

pour $q' \neq q'_1$ et $q' \neq q'_2$. Par construction, on a :

$$[\mu(a)]_{q_1 q'_1} = [\mu(a)]_{q_1 q'_2} = [\mu(a)]_{q_2 q'_1} = [\mu(a)]_{q_2 q'_2} = \tau,$$

et tous les autres coefficients de $\mu(a)$ égaux à ϵ . On a alors

$$[x_{G_1||G_2}(wa)]_{q'_1} = \tau + \max([x_{G_1||G_2}(w)]_{q_1}, [x_{G_1||G_2}(w)]_{q_2}), \quad (6)$$

$$[x_{G_1||G_2}(wa)]_{q'_2} = \tau + \max([x_{G_1||G_2}(w)]_{q_1}, [x_{G_1||G_2}(w)]_{q_2}), \quad (7)$$

pour $a^\bullet = \{q'_1, q'_2\}$ avec $q'_1 \in \mathcal{Q}_1$, $q'_2 \in \mathcal{Q}_2$, et

$$[x_{G_1||G_2}(wa)]_{q'} = \epsilon \quad (8)$$

pour $q' \neq q'_1$ et $q' \neq q'_2$. L'analogie entre les équations (3)-(5) et (6)-(8) permet de conclure à l'égalité $x_{\mathcal{G}_1||\mathcal{G}_2}(wa) = x_{G_1||G_2}(wa)$.

Considérons une transition correspondant à un événement privé de G_1 ou G_2 , soit $a \in A_1 \setminus A_2$. Elle peut être franchie si sa place d'entrée $q_1 = \bullet a$, $q_1 \in \mathcal{P}_1$, contient un jeton à l'issue de la séquence w , arrivé à l'instant $[x_{\mathcal{G}_1||\mathcal{G}_2}(w)]_{q_1}$. Notons $q'_1 = a^\bullet$, $q'_1 \in \mathcal{P}_1$, on a :

$$[x_{\mathcal{G}_1||\mathcal{G}_2}(wa)]_{q'_1} = \tau + [x_{\mathcal{G}_1||\mathcal{G}_2}(w)]_{q_1}, \quad (9)$$

et

$$[x_{\mathcal{G}_1||\mathcal{G}_2}(wa)]_q = \epsilon \quad (10)$$

pour $q \neq q'_1$ et $q \in \mathcal{Q}_1$. Aucune transition n'est alors franchie dans \mathcal{G}_2 et

$$[x_{\mathcal{G}_1||\mathcal{G}_2}(wa)]_q = [x_{\mathcal{G}_1||\mathcal{G}_2}(w)]_q \text{ pour } q \in \mathcal{P}_2. \quad (11)$$

Par construction, on a :

$$[\mu(a)]_{q_1 q'_1} = \tau, [\mu(a)]_{qq} = \epsilon \text{ pour } q \in \mathcal{Q}_2.$$

et tous les autres coefficients de $\mu(a)$ égaux à ϵ . On a alors

$$[x_{G_1||G_2}(wa)]_{q'_1} = \tau + [x_{G_1||G_2}(w)]_{q_1}, \quad (12)$$

et

$$[x_{G_1||G_2}(wa)]_q = \epsilon \text{ pour } q \in \mathcal{Q}_1 \text{ et } q \neq q'_1 \quad (13)$$

$$[x_{G_1||G_2}(wa)]_q = [x_{G_1||G_2}(w)]_q \text{ pour } q \in \mathcal{Q}_2. \quad (14)$$

L'analogie entre les équations (9)-(11) et (12)-(14) permet de conclure à nouveau à l'égalité $x_{\mathcal{G}_1||\mathcal{G}_2}(wa) = x_{G_1||G_2}(wa)$.

Exemple 6. Considérons les automates G_1 et G_2 (resp., les graphes d'états \mathcal{G}_1 et \mathcal{G}_2) représentés sur les figures 1 et 3 (resp., les figures 2 et 3) dont le produit synchrone a été présenté dans l'exemple 5. On a :

$$\begin{aligned} x_{G_1||G_2}(accbda) &= \alpha \mu(a) \mu(c) \mu(c) \mu(b) \mu(d) \mu(a) \\ &= (\epsilon \ 11 \ \epsilon \ 11) \end{aligned}$$

Le réseau de Petri de la figure 5 correspond à la composition de \mathcal{G}_1 et \mathcal{G}_2 en fusionnant la transition commune a . En considérant que l'évolution débute à l'instant 0, la séquence de franchissements $accbda$ s'achève bien dans ce réseau à l'instant 11, si bien que les dates d'arrivée des jetons dans les places sont bien obtenues *via* $x_{G_1||G_2}(accbda)$.

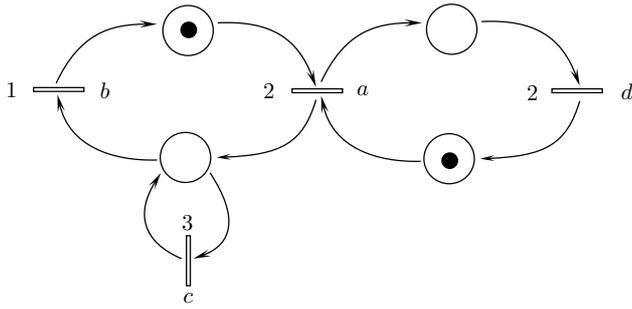


FIGURE 5. Réseau de Petri $G_1||G_2$

5. CONCLUSION

Un produit synchrone est proposé pour les automates $(\max, +)$. Il permet d'une part d'adopter une démarche modulaire pour la modélisation des systèmes à événements discrets, et d'autre part d'envisager une commande modulaire pour les automates $(\max, +)$ dans la continuité des travaux Komenda et al. (2009b). Ce produit synchrone a la caractéristique remarquable de fournir un automate dont le nombre d'états est égal à la somme du nombre des états des automates synchronisés entre eux.

RÉFÉRENCES

- Baccelli, F., Cohen, G., Olsder, G.J., and Quadrat, J.P. (1992). *Synchronization and Linearity*. Wiley.
- Buchholz, P. and Kemper, P. (2003). Weak bisimulation for $(\max/+)$ automata and related models. *J. Autom. Lang. Comb.*, 8, 187–218.
- Cassandras, C.G. and Lafortune, S. (2006). *Introduction to Discrete Event Systems*. Springer-Verlag New York, Inc.
- David, R. and Alla, H. (1989). *Du Grafset aux réseaux de Petri*. Hermès, Paris.
- Gaubert, S. (1995). Performance Evaluation of $(\max,+)$ Automata. *IEEE Transaction on Automatic Control*, vol. 40(12), 2014–2025.
- Gaubert, S. and Mairesse, J. (1999). Modeling and Analysis of Timed Petri Nets using Heaps of Pieces. *IEEE Transaction on Automatic Control*, vol. 44(4), 683–698.
- Houssin, L. (2011). Cyclic jobshop problem and $(\max,+)$ algebra. In *18th IFAC World Congress*, 2717–2721. Milan, Italy.
- Kirsten, D. and Lombardy, S. (2009). Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata. In *26th International Symposium on Theoretical Aspects of Computer Science (STACS 2009)*, 589–600.
- Klimann, I., Lombardy, S., Mairesse, J., and Prieur, C. (2004). Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. *Theor. Comput. Sci.*, 327.
- Komenda, J., Lahaye, S., and Boimond, J.L. (2009a). Le produit synchrone des automates $(\max,+)$. *Special issue of Journal Européen des Systèmes Automatisés (JESA)*, vol. 43(7), 1033–1047.
- Komenda, J., Lahaye, S., and Boimond, J.L. (2009b). Supervisory Control of $(\max,+)$ Automata : A Behavioral Approach. *Discrete Event Dynamic Systems*, vol. 19(4), 525–549.
- Lombardy, S. and Sakarovitch, J. (2006). Sequential? *Theoretical Computer Science*, 359(1-2), 224–244.
- Ramadge, P.J.G. and Wonham, W.M. (1989). The control of discrete event systems. *Proceedings of The IEEE*, 77, 81–98.
- Su, R., van Schuppen, J., and Rooda, J. (2012). The synthesis of time optimal supervisors by using heaps-of-pieces. *IEEE Transaction on Automatic Control*, vol. 57(1), 105–118.
- Su, R. and Woeginger, G.J. (2011). String execution time for finite languages : Max is easy, min is hard. *Automatica*, 47(10), 2326–2329.