HAL
open science

Contributions à l'étude des systèmes à événements discrets à partir de modèles définis sur des semi-anneaux idempotents

Sébastien Lahaye

▶ **To cite this version:**

Sébastien Lahaye. Contributions à l'étude des systèmes à événements discrets à partir de modèles définis sur des semi-anneaux idempotents. Sciences de l'ingénieur [physics]. Université d'Angers, 2011. tel-00841440

HAL Id: tel-00841440

https://theses.hal.science/tel-00841440

Submitted on 4 Jul 2013

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

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

Université d'Angers

Habilitation à diriger des recherches

Contributions à l'étude des systèmes à événements discrets à partir de modèles définis sur des semi-anneaux idempotents

LAHAYE Sébastien
sebastien.lahaye@univ-angers.fr

Soutenue le 21 Novembre 2011 devant le jury suivant :

Hassane ALLA	Rapporteur	Professeur, Université Joseph Fourier
Stéphane GAUBERT	Rapporteur	Directeur de recherche, INRIA
Jean-Claude HENNET	Rapporteur	Directeur de recherche, CNRS
Jean-Louis BOIMOND	Examinateur	Professeur, Université d'Angers
Jean-Jacques LESAGE	Examinateur	Professeur, ENS de Cachan

Laboratoire d'Ingénierie des Systèmes Automatisés - LISA EA 4094

Table des matières

Introduction **1**

1 Présentation synthétique **3**
1.1 Curriculum Vitae . . . 3
1.1.1 Titres universitaires . . . 3
1.1.2 Fonctions occupées . . . 3
1.1.3 Activités pédagogiques . . . 4
1.1.4 Activités administratives . . . 4
1.2 Activités de recherche . . . 4
1.2.1 Thèmes abordés . . . 4
1.2.2 Encadrement doctoral et scientifique . . . 5
1.2.3 Animation Scientifique . . . 6
1.2.4 Production scientifique . . . 7

2 Contributions à l'étude des systèmes à événements discrets **13**
2.1 Modélisation des systèmes à événements discrets dans l'algèbre des dioïdes . . . 14
2.1.1 Système à événements discrets . . . 14
2.1.2 Modélisation du comportement temporisé dans l'algèbre des dioïdes . . . 15
2.1.3 Systèmes linéaires dans l'algèbre des dioïdes . . . 18
2.2 Des travaux visant à étendre les classes des systèmes étudiés dans l'algèbre des dioïdes . . . 19
2.2.1 Automates $(\max, +)$ . . . 20
2.2.2 Modélisation des systèmes à événements discrets à l'aide d'automates $(\max, +)$ . . . 22
2.2.3 Une approche de commande des automates $(\max, +)$ . . . 34
2.2.4 Conclusions et perspectives . . . 40
2.3 Des applications aux systèmes de transport . . . 41
2.3.1 Réseaux de transport . . . 41
2.3.2 Trafic routier . . . 45
2.3.3 Conclusion et perspectives . . . 46

3 Conclusions et perspectives **49**
3.1 Conclusions . . . 49
3.2 Perspectives . . . 49

Bibliographie **50**

Annexes **54**

Table des figures

2.1 Configurations au sein d'un réseau de Petri modélisant les phénomènes identifiés au sein d'un système à événements discrets. 16
2.2 Un automate (max,+). 22
2.3 Un réseau de Petri 1-sauf 24
2.4 Automate (max,+) dont le comportement reconnaît l'évolution du réseau de Petri représenté sur la figure 2.3 24
2.5 Illustration de l'application qui a une transition d'état d'un automate fait correspondre une transition dans un réseau de Petri. 26
2.6 Deux automates $(\max, +)$ et leurs graphes d'état équivalents. 28
2.7 Automate (max,+) $\mathfrak{G}$. 30
2.8 Interaction du superviseur G_s avec le système G sous la forme d'un retour d'information. 35
2.9 Automate (max,+) modélisant le système. 39

Introduction

Ce manuscrit a été rédigé en vue d'obtenir l'habilitation à diriger des recherches. J'y présente mon implication dans l'enseignement supérieur et la recherche au sein de l'Université d'Angers depuis mon recrutement en tant que maître de conférences, et plus précisément en tant qu'enseignant à l'ISTIA [1] et en tant que chercheur au LISA [2].

Au LISA, dirigé par Jean-Louis Ferrier au moment de mon recrutement, j'ai intégré l'équipe *Modèles et Systèmes Dynamiques*. Mon travail de recherche a profité des interactions avec les différents membres de cette équipe, et en particulier avec Jean-Louis Boimond (comme le laisseront apparaître les références dans la suite de ce manuscrit). Il concerne le comportement temporisé des systèmes à événements discrets en utilisant des modèles définis sur une structure algébrique de semi-anneau idempotent, encore appelée dioïde.

Le document est structuré de la manière suivante.

- Le premier chapitre présente mon *curriculum vitae* ainsi qu'un survol général de mes activités de recherche.
- Le deuxième chapitre fait une synthèse plus détaillée de travaux motivés par deux préoccupations principales durant ces onze dernières années : élargir la classe des modèles relevant de la théorie des systèmes sur l'algèbre des dioïdes ; appliquer les résultats de la théorie des systèmes sur les dioïdes à l'analyse des systèmes de transport. En annexe, trois publications sont jointes pour que le lecteur puisse trouver plus de détails sur ces travaux.
- Dans le troisième et dernier chapitre, je tire un bilan et envisage des perspectives à mes travaux.

1. Institut des Sciences et Techniques de l'Ingénieur d'Angers, école d'ingénieurs de l'Université d'Angers.
2. Laboratoire d'Ingénierie des Systèmes Automatisés, EA 4094.

Chapitre 1

Présentation synthétique

1.1 Curriculum Vitae

Sébastien Lahaye
Né le 23 mars 1973 à Cholet (49)
Vie maritale, 4 enfants

ISTIA - LISA
62, avenue Notre-Dame du Lac
49000 Angers
02 41 22 65 65
`sebastien.lahaye@univ-angers.fr`

1.1.1 Titres universitaires

2000 Doctorat de l'Université d'Angers, spécialité Automatique et Informatique Appliquée, mention très honorable avec les félicitations du jury
Mémoire intitulé *"Contribution à l'étude des systèmes linéaires non stationnaires dans l'algèbre des dioïdes"*
Préparé sous la direction de Jean-Louis Boimond et soutenu le 27 Octobre 2000 devant le jury suivant : Christian Commault (rapporteur), Abdellah El Moudni (rapporteur), Geert Jan Olsder (rapporteur), Guy Juanole (président), Edouard Wagneur, Jean-Louis Ferrier, Jean-Louis Boimond.

1996 D.E.A. de l'Ecole Centrale de Nantes/Université de Nantes, spécialité Automatique et Informatique Appliquée, mention assez bien.

1.1.2 Fonctions occupées

Depuis 2001 Maître de conférences : enseignant à l'Institut des Sciences et Techniques de l'Ingénieur d'Angers (ISTIA), école d'ingénieurs de l'Université d'Angers ; chercheur au Laboratoire d'Ingénierie des Systèmes Automatisés (LISA), EA 4094.

2000 - 2001 Attaché temporaire (ATER) : enseignant à l'ISTIA, chercheur au LISA.

1997 - 2000 Doctorant au LISA ; **moniteur** à l'ISTIA.

1.1.3 Activités pédagogiques

Depuis mon recrutement, mes enseignements relèvent de l'automatique et du génie informatique : automatique, identification/estimation, automatismes industriels, supervision industrielle, administration de réseaux locaux, langage Perl... Ma charge a été en moyenne de 250 h. eq. TD par an et je suis intervenu très majoritairement au niveau master/cycle ingénieur à l'ISTIA.

1.1.4 Activités administratives

Depuis 2007 Responsable de la filière et de l'option Automatique et Génie Informatique de l'ISTIA, école d'ingénieurs de l'Université d'Angers
[`http://www.istia.univ-angers.fr/index.php?id=agi`].
L'équipe pédagogique de la filière se compose en 2010-2011 de : 4 Pr, 9 MCF, 3 ATER et environ 20 vacataires.
L'option, qui correspond aux deux dernières années du cycle ingénieur à l'ISTIA, est suivie en 2010-2011 par 96 élèves ingénieurs.

2007 - 2011 Membre élu du conseil d'administration de l'ISTIA.

Depuis 2003 Membre élu de la commission de spécialistes/du comité consultatif 60-63-ièmes sections de l'Université d'Angers.

2002-2011 Initiateur et responsable de l'action Analyse des réseaux de transport au sein du LISA à laquelle participent en 2011 : J.-L. Boimond (Pr), S. Lahaye (MCF) et J. Rousseau (ATER)

2003 - 2007 Membre élu du conseil de laboratoire du LISA.

2002 - 2007 Responsable logistique au sein de la filière Automatique et Génie Informatique.

1.2 Activités de recherche

Les travaux de recherche décrits ci-après ont été réalisés depuis mon recrutement en tant que maître de conférences au sein de l'équipe *Modèles et Systèmes Dynamiques* (MSD) du LISA. Leur thématique, détaillée dans la section qui suit, m'est familière depuis 1997 puisque j'ai préparé ma thèse de doctorat dans cette même équipe. Les travaux sont ensuite déclinés en termes d'encadrement et d'animation scientifiques. Un bilan de la production clôt cette section.

1.2.1 Thèmes abordés

Mon travail de recherche s'intéresse aux *systèmes à événements discrets* pour lesquels différentes approches ont été développées depuis une quarantaine d'années. Les *langages et automates* et les *réseaux de Petri* sont les "paradigmes" les plus connus pour la modélisation des phénomènes inhérents aux systèmes à événements discrets. Oscillant entre ces deux théories, mes travaux ont pour caractéristiques :

- de se focaliser sur le comportement temporisé des systèmes à événements discrets,
- d'utiliser des modèles définis sur des structures algébriques de *semi-anneaux idempotents*, souvent appelées *dioïdes*,

- de privilégier les analogies avec la théorie classique des systèmes.

Deux principaux objectifs ont été poursuivis[1], à savoir :

- *Elargir la classe des modèles relevant de la théorie des systèmes sur les dioïdes* (encadrements [T2,T4] et collaboration [C1,C2] détaillés ci-après).

- *Appliquer les résultats de la théorie des systèmes sur les dioïdes à l'analyse des systèmes de transport* (encadrements [M1,T1,M3,T3]).

1.2.2 Encadrement doctoral et scientifique

J'ai eu la chance de pouvoir (co-)encadrer sept étudiants lors de leur formation scientifique.

Laurent Houssin a obtenu

[M1] le DEA de l'Ecole Centrale de Nantes - Université de Nantes, spécialité Automatique et Informatique Appliquée, en 2003 ;

[T1] le doctorat de l'Université d'Angers, spécialité Automatique et Informatique, en 2006 (co-encadrant : J.L. Boimond à 30% ; rapporteurs : J.-P. Quadrat, J.-J. Loiseau)

Les travaux avec Laurent ont eu pour objet principal l'étude des réseaux de transport dans le cadre de la théorie des systèmes $(\max, +)$ linéaires. Une commande en juste-à-temps sous contraintes temporelles a été proposée. Celle-ci peut s'appliquer à la génération des tables d'horaires pour les réseaux de bus [ACL3], [ACLN3, ACLN4], [ACTI22, ACTI11, ACTI13], [ACTN3]. On a également abordé la synthèse de correcteurs de type retour de sortie minimisant les retards. Ces travaux ont été motivés par leur application potentielle à la régulation des véhicules en amont d'un noeud ferroviaire [ACL2] [ACTI12].
Laurent est maître de conférences à l'université Paul Sabatier de Toulouse-LAAS depuis 2007.

Michel Al Saba a préparé à partir de 2003 puis obtenu

[T2] le doctorat de l'Université d'Angers, spécialité Automatique et Informatique, en 2006 (co-encadrant : J.L. Boimond à 50% ; rapporteurs : C. Commault, J.-J. Loiseau).

Dans ses travaux, Michel a notamment porté son attention sur les *modèles de type tas* introduits et étudiés par Stéphane Gaubert et Jean Mairesse [Gaubert and Mairesse, 1999]. Il s'agit d'automates à pondération dans l'algèbre $(\max, +)$ [Gaubert, 1995] qui permettent aisément de modèliser des phénomènes de choix (alors que la prise en compte de ces phénomènes dans les dioïdes a toujours conduit à des modèles difficilement exploitables [Libeaut, 1996, Gaujal, 1994]). La modélisation des systèmes flexibles de production manufacturière a ainsi pu être abordée, et différentes techniques de commande ont été proposées [ACLN1], [ACTI1, ACTI2, ACTI3].
Michel est enseignant-chercheur à Damas (Syrie).

Aurélien Correïa a obtenu

[M2] le DEA de l'Ecole Centrale de Nantes - Université de Nantes, spécialité Automatique et Informatique Appliquée, en 2004 (co-encadrant : B. Cottenceau à 50%)

1. En cohérence avec ceux des actions auxquelles j'ai contribué au sein de l'équipe MSD du LISA.

Son travail a consisté en l'élaboration d'une commande pour les systèmes à événements discrets dits *à temps critique* ou *à temps contraint* [ACTI21, ACTI6].
Aurélien a ensuite obtenu en 2007 un doctorat de l'Université Technique de Belfort-Montbéliard, et est depuis 2008 ingénieur de recherche (Hitachi SAS, Royaume-Uni).

Julien Rousseau a préparé à partir de 2006 puis obtenu

[T3] le doctorat de l'Université d'Angers, spécialité Automatique et Informatique, en 2010 (co-encadrants : C. Martinez à 35%, J.L. Boimond à 30% ; rapporteurs : J.-C. Hennet, N. Sauer).

D'un point de vue théorique, les travaux de Julien ont essentiellement contribué aux techniques d'identification, et plus précisément d'estimation, des systèmes linéaires sur les dioïdes. A partir des mesures usuelles du trafic, ces travaux peuvent s'appliquer pour obtenir un modèle du trafic routier sur un tronçon de route. La modélisation et la simulation d'infrastructures routières ont subséquemment pu être abordées [ACTI23].
Julien est actuellement ATER à l'IUT d'Angers.

Tony Doat a obtenu

[M3] le master recherche de l'Université d'Angers, spécialité Systèmes Dynamiques et Signaux, en 2008.

Dans le cadre d'un partenariat avec la société Artefacto située à Rennes, le travail de Tony a eu pour objet de synthétiser des modèles représentant de façon réaliste la dynamique des tramways sur la ligne construite à Angers. Leur implémentation a donné lieu à un simulateur s'interfaçant avec le démonstrateur 3D de la société Artefacto.
Tony est depuis 2008 doctorant au Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur (LIMSI, Orsay).

Rabah Boukra prépare depuis 2010

[T4] le doctorat de l'Université d'Angers, spécialité Automatique et Informatique (co-encadrant : J.L. Boimond à 30%).

Avec Rabah, nous envisageons la commande sous observation partielle ainsi que la commande décentralisée des automates $(\max, +)$.

1.2.3 Animation Scientifique

[C1] Depuis 2007, je collabore avec Jan Komenda (Académie tchèque des sciences) sur la commande des automates $(\max, +)$.

Jean-Louis Boimond est également impliqué. Le sujet de cette collaboration fait suite aux travaux de thèse de Michel Al Saba [T2]. Elle a pris la forme :

- d'un partenariat Hubert Curien (Barrande) sur la période 2007-2008 (séjours à Brno : 1 semaine en 2007, 1 semaine en 2008 ; accueils du partenaire : 2 semaines en 2007, 2 semaines en 2008) ;
- de séjours invités (accueil de Jan pendant 3 semaines en 2009, puis 2 semaines en 2010 ; séjour d'une semaine en mai 2010, puis d'une semaine en mai 2011).

La collaboration a donné lieu aux publications [OS3], [ACL4], [ACLN5], [ACTI14, ACTI15, ACTI16, ACTI17, ACTI18, ACTI19], [ACTN4].

[C2] En 2010, une collaboration avec Pavel Spacek (Université Technique de Brno) a débuté sur l'analyse des réseaux de Petri temporels.

Cette collaboration, à laquelle participe également Jan Komenda, a donné lieu à l'accueil de Pavel pendant deux semaines en 2010 et a débouché sur la communication [ACTI20].

J'ai séjourné une semaine en 2009 (en compagnie de J.-L. Boimond) aux Pays-Bas, Delft University of Technology, à l'initiative de Geert-Jan Olsder. Ce séjour fut l'occasion d'exposés croisés et d'échanges en vue de démarrer une collaboration avec les membres du Delft Center for Systems and Control.

J'ai été membre du comité d'organisation des événements scientifiques suivants :

- Journées Démonstrateurs du Club EAA, novembre 2010, Angers ;
- JD-JN-MACS 2009, journées nationales et doctorales du GDR MACS, mars 2009, Angers ;
- ICINCO 2007, mai 2007, Angers ;
- Journées Démonstrateurs du Club EAA, mars 2006, Angers.

Je suis sollicité régulièrement pour évaluer des articles et des communications. Pour la période courant depuis début 2010 : *Automatica* (2 articles en 2010, 1 en 2011), *Linear Algebra and Applications* (1 article en 2010), *Asian Journal of Control* (2 articles en 2010), *Conference on Decision and Control* (1 communication en 2010, 1 en 2011), *IFAC World Congress* (2 communications en 2011), *Symposium on Mathematical Theory of Networks and Systems* (1 communication en 2010).

1.2.4 Production scientifique

Outre les références précisées dans le cadre des encadrements ou collaborations, le lecteur trouvera dans la bibliographie qui suit :

- des publications issues de travaux dans le prolongement de ma thèse sur les systèmes (max,+) linéaires non stationnaires [ACL7, ACL6, ACL5] ;
- des publications issues d'une collaboration informelle et restreinte aux travaux de thèse de S. Hamaci portant sur la linéarisation des graphes d'événements discrets avec multiplieurs et les modèles hybrides [OS1, OS2], [ACL1], [ACLN2], [ACTI7, ACTI8, ACTI9, ACTI10], [ACTN2] ;
- des publications résultant de travaux avec Jean-Louis Boimond sur la synthèse de correcteurs pour les systèmes linéaires sur les dioïdes [ACTI4, ACTI5], [ACTN1].

Ouvrages et chapitres d'ouvrage

[OS1] S. Hamaci, J.-L. Boimond, and S. Lahaye. *Part 3-Signal Processing, Systems Modeling and Control. On Modeling and Control of Discrete Timed Event Graphs with Multipliers using (Min,+) Algebra.* SPRINGER, 2006. J. Bras et al. (Editors), Informatics in Control, Automation and Robotics I. ISBN 1-4020-4136-5.

[OS2] S. Hamaci, J.-L. Boimond, and S. Lahaye. *Part 3-Signal Processing, Systems Modeling and Control. Performance Analysis of Timed Event Graphs with Multipliers using (Min, +) Algebra.* SPRINGER, 2007. J. Filipe et al. (Editors), Informatics in Control, Automation and Robotics II. ISBN 1-4020-5625-3.

[OS3] J. Komenda, J.-L. Boimond, and S. Lahaye. *New Developments in Robotics, Automation and Control.* INTECH EDUCATION AND PUBLISHING, 2008. Chapter : Heap Models, Composition and Control p. 427-448 ISBN : 978-953-7619-20-6.

Articles répertoriés dans *ISI Web of Knowledge*

[ACL1] S. Hamaci, J.-L. Boimond, and S. Lahaye. Modeling and Control of Hybrid Timed Event Graphs with Multipliers using (Min,+) Algebra. *Discrete Event Dynamic Systems*, vol. 16(2) :241–256, 2006.

[ACL2] L. Houssin, S. Lahaye, and J.-L. Boimond. Control of (max,+)-linear systems minimizing delays. *Discrete Event Dynamic Systems.* accepté sous la réserve de modifications mineure en cours.

[ACL3] L. Houssin, S. Lahaye, and J.-L. Boimond. Just in Time Control of constrained (max,+)-Linear Systems. *Discrete Event Dynamic Systems*, vol. 17(2) :159–178, 2007.

[ACL4] J. Komenda, S. Lahaye, and J.-L. Boimond. Supervisory Control of (max,+) Automata : A Behavioral Approach. *Discrete Event Dynamic Systems*, vol. 19(4) :525–549, 2009.

[ACL5] S. Lahaye, J.-L. Boimond, and J.-L. Ferrier. Just in Time Control of Time-Varying Discrete Event Dynamic Systems in (max,+) Algebra. *International Journal of Production Research (IJPR)*, vol. 46(19) :5337–5348, 2008.

[ACL6] S. Lahaye, J. L. Boimond, and L. Hardouin. Linear Periodic Systems Over Dioids. *Discrete Event Dynamic Systems*, vol. 14 :133–152, 2004.

[ACL7] S. Lahaye, L. Hardouin, and J.-L. Boimond. Models Combination in (Max,+) Algebra for the Implementation of a Simulation and Analysis Software. *Kybernetika*, vol. 39(2) :143–154, 2003.

Autres articles dans des journaux avec comité de lecture

[ACLN1] M. Alsaba, J.-L. Boimond, and S. Lahaye. Sur la commande des systèmes flexibles de production manufacturière par l'algèbre des dioïdes. *Sciences et Technologies de l'Automatique (e-STA)*, vol. 4(2), 2007.

[ACLN2] S. Hamaci, J.-L. Boimond, S. Lahaye, and M. Mostefaoui. Sur la linéarisation des graphes d'événements temporisés discrets avec multiplieurs dans l'algèbre (min,+). *Sciences et Technologies de l'Automatique (e-STA)*, vol. 1(3), 2004.

[ACLN3] L. Houssin, S. Lahaye, and J.-L. Boimond. Modélisation et commande de réseaux de bus urbains dans l'algèbre des dioïdes. *Sciences et Technologies de l'Automatique (e-STA)*, vol. 1(3), 2004.

[ACLN4] L. Houssin, S. Lahaye, and J.-L. Boimond. Commande en juste-à-temps sous contraintes de systèmes (max,+)-linéaires. *Special issue of Journal Européen des Systèmes Automatisés (JESA)*, 39 :335–350, 2005.

[ACLN5] J. Komenda, S. Lahaye, and J.-L. Boimond. Le produit synchrone des automates (max,+). *Special issue of Journal Européen des Systèmes Automatisés (JESA)*, vol. 43(7) :1033–1047, 2009.

Communications avec actes dans un congrès international

[ACTI1] M. Alsaba, J-L. Boimond, and S. Lahaye. On just in time control of flexible manufacturing systems via dioid algebra. In *Information Control Problems in Manufacturing (INCOM 2006)*, volume 2, pages 137–142, Saint-Etienne, France, Mai 2006.

[ACTI2] M. Alsaba, J.-L. Boimond, and S. Lahaye. Sur la commande des systèmes flexibles de production manufacturière par l'algèbre des dioïdes. In *Conférence Internationale Francophone d'Automatique (CIFA 2006)*, Bordeaux, France, Mai 2006.

[ACTI3] M. Alsaba, S. Lahaye, and J.-L. Boimond. On just in time control of switching max-plus linear systems. In *International Conference on Informatics in Control, Automation and Robotics(ICINCO 2006)*, Setùbal, Portugal, Août 2006.

[ACTI4] J.-L. Boimond and S. Lahaye. On Steady State Controller in Min-Plus Algebra. In *15th conf. International Liner Algebra Society (ILAS 2008)*, Cancun, Mexico, Juin 2008.

[ACTI5] J.-L. Boimond and S. Lahaye. On Proportional Controller in (Min,+) Algebra. In *Workshop on Discrete Event Systems (WODES 2010)*, Berlin, Allemagne, Août 2010.

[ACTI6] A. Correïa, S. Lahaye, and B. Cottenceau. Commande de graphes d'événements temporisés partiellement commandables avec contraintes temporelles. In *Conférence Internationale Francophone d'Automatique (CIFA 2006)*, Bordeaux, France, Mai 2006.

[ACTI7] S. Hamaci, J-L . Boimond, and S. Lahaye. Performance analysis of timed event graphs with multipliers using (min, +) algebra. In *IFAC, International Conference on Informatics in Control, Automation and Robotics 2005*, volume 4, Barcelone, Espagne, Septembre 2005.

[ACTI8] S. Hamaci, J.-L. Boimond, and S. Lahaye. Modeling and Control of Discrete Timed Event Graphs with Multipliers using (Min,+) Algebra. In *International Conference on Informatics in Control, Automation and Robotics (ICINCO 2004)*, volume 3, pages 32–37, Setùbal, Portugal, Août 2004.

[ACTI9] S. Hamaci, J.-L. Boimond, and S. Lahaye. On Hybrid Timed Event Graphs with Multipliers in Dioid. In *16th International Conference on Control Systems and Computer Science (CSCS 2006)*, Bucharest, Romania, Mai 2007.

[ACTI10] S. Hamaci, J.-L. Boimond, S. Lahaye, and M. Mostefaoui. On the Linearizability of Discrete Timed Event Graphs with Multipliers using (Min,+) Algebra. In *Workshop on Discrete Event Systems (WODES'2004)*, pages 367–372, Reims, France, Septembre 2004.

[ACTI11] L. Houssin, S. Lahaye, and J.-L. Boimond. Modelling and Control of Urban Bus Networks in Dioids Algebra. In *Workshop on Discrete Event Systems (WODES'2004)*, Reims, France, Septembre 2004.

[ACTI12] L. Houssin, S. Lahaye, and J.-L. Boimond. Control of Constrained (max,+)-Linear Systems Minimizing Delays. In *Information Control Problems in Manufacturing (INCOM 2006)*, Saint-Etienne, France, Juin 2006.

[ACTI13] L. Houssin, S. Lahaye, and J.-L. Boimond. Timetable Synthesis Using (max,+) Algebra. In *Information Control Problems in Manufacturing (INCOM 2006)*, Saint-Etienne, France, Juin 2006.

[ACTI14] J. Komenda, J.-L. Boimond, and S. Lahaye. Supervisory Control of Heap Models using Synchronous Composition. In *International Conference on Informatics in Control, Automation and Robotics(ICINCO 2007)*, pages 467–475, Angers, France, Mai 2007. Special Session : From Planning to Control of Manufacturing Systems.

[ACTI15] J. Komenda, J.-L. Boimond, and S. Lahaye. Control of (max,+) automata : logical and timing aspects. In *9th international workshop on discrete event systems (WODES 2008)*, pages 55–60, Göteborg, sweden, Mai 2008.

[ACTI16] J. Komenda, S. Lahaye, and J.-L. Boimond. Controllability of (max,+) Formal Power Series. In *IFAC Workshop on Dependable Control of Discrete Systems (DCDS 2009)*, Bari, Italy, Juin 2009.

[ACTI17] J. Komenda, S. Lahaye, and J.-L. Boimond. Supervisory Control of (max,+) Automata : A Single Step Approach. In *European Control Conference (ECC'09)*, Budapest, Hongrie, Août 2009.

[ACTI18] J. Komenda, S. Lahaye, and J.-L. Boimond. Commande supervisée des automates (max,+). In *Conférence Internationale Francophone d'Automatique (CIFA 2010)*, Nancy, France, Juin 2010.

[ACTI19] J. Komenda, S. Lahaye, and J.-L. Boimond. Synchronous Composition of Interval Weighted Automata. In *International Workshop on Discrete Event Systems (WODES 2010)*, Berlin, Allemagne, Août 2010.

[ACTI20] J. Komenda, P. Spacek, and S. Lahaye. Application of product dioids for dead token detection in interval P-time event graphs. In *IFAC World Congress*, Milan, Italie, Août 2011.

[ACTI21] S. Lahaye, B. Cottenceau, and A. Correïa. Commande de graphes d'événements temporisés avec contrainte de temps critique. In *Conférence Internationale Francophone d'Automatique (CIFA 2004)*, Douz, Tunisie, Novembre 2004.

[ACTI22] S. Lahaye, L. Houssin, and J.-L. Boimond. Modelling of Urban Bus Networks in Dioids Algebra. In *first International Symposium on Positive Systems : Theory and Applications (POSTA 2003)*, pages 23–30, Rome, Italie, Août 2003. Lecture Notes in Control and Information Sciences, Ed. L. BENVENUTI, A. DE SANTIS and L. FARINA, Springer-Verlag.

[ACTI23] J. Rousseau, S. Lahaye, C. Martinez, and J.-L. Boimond. A min-plus approach for traffic flow modelling. In *International Conference on Informatics in Control, Automation and Robotics (ICINCO 2009)*, Milan, Italie, Juillet 2009.

Communications avec actes dans un congrès national

[ACTN1] J.-L. Boimond and S. Lahaye. Sur la précision asymptotique des systèmes (min, +) linéaires continus. In *Modélisation des Systèmes Réactifs (MSR 2011)*, Lille, France, 2011.

[ACTN2] S. Hamaci, J.-L. Boimond, and S. Lahaye. Sur le comportement dynamique des réseaux de Petri valués dans l'algèbre des dioïdes. In *Conception et Production Intégrées (CPI 2007)*, Rabat, Maroc, Octobre 2007.

[ACTN3] L. Houssin, S. Lahaye, and J.-L. Boimond. Commande en juste-à-temps sous contraintes de systèmes (max,+)-linéaires. In *Modélisation des Systèmes Réactifs (MSR 2005)*, volume 39, pages 335–350, Grenoble, France, Octobre 2005. Publié dans une édition spéciale de Journal Européen des Systèmes Automatisés (JESA).

[ACTN4] J. Komenda, S. Lahaye, and J.-L. Boimond. Le produit synchrone des automates (max,+). In *Modélisation des Systèmes Réactifs (MSR 2009)*, volume 43/7-9, pages 1033–1047, Nantes, France, Novembre 2009. Publié dans une édition spéciale de Journal Européen des Systèmes Automatisés (JESA).

Chapitre 2

Contributions à l'étude des systèmes à événements discrets

L'évolution de nombreux systèmes est en partie régie par des règles opérationnelles, ou algorithmes, qui répondent à des faits ponctuels. Par exemple, au sein d'un système automatisé, l'assemblage de deux pièces peut être déclenché au moment où les pièces ont été acheminées sur la machine dédiée. Les départs de véhicule en correspondance au sein d'un réseau de transport sont autorisés au fil des arrivées à destination. Le transfert d'un fichier sur un réseau informatique n'est achevé que lorsque l'ensemble des paquets correspondant ont été réceptionnés. Lorsque l'on s'intéresse exclusivement à de tels aspects de la dynamique d'un système, celui-ci est qualifié de *système à événements discrets* (pour signifier que l'évolution est régie par l'occurrence d'événements).

Plusieurs modèles ont été proposés pour étudier spécifiquement les systèmes à événements discrets. Les études présentées dans ce chapitre font appel à des modèles reposant sur la structure algébrique de semi-anneau idempotent, encore appelée *dioïde*. Les apports de cette algèbre pour l'étude des systèmes à événements discrets ont été mis à jour au début des années 80 [Cohen et al., 1984] en partant du constat qu'une partie des phénomènes survenant dans ces systèmes peuvent être décrits par des équations linéaires dans un dioïde. Depuis, de nombreux travaux ont contribué à construire une véritable *théorie des systèmes sur les dioïdes*. Citons les ouvrages collectifs [Baccelli et al., 1992, Heidergott et al., 2006] et les premières thèses [Moller, 1988, Gaubert, 1992, Braker, 1993].

Ce chapitre est une synthèse d'une partie des travaux sur cette thématique auxquels je participe depuis une dizaine d'années. Il comporte trois parties.

1. On commence par préciser dans la section 2.1 un certain nombre de définitions et concepts permettant de comprendre comment un système à événements discrets peut être modélisé dans l'algèbre des dioïdes.

2. On résume ensuite dans la section 2.2 des travaux explorant des modèles (reposant sur l'algèbre des dioïdes) qui permettent d'étudier une classe de systèmes à événements discrets plus large que celle considérée dans les premières études (citées ci-dessus). On détaillera en particulier des travaux récents portant sur les automates $(\max, +)$.

3. La dernière partie, c'est-à-dire la section 2.3, est consacrée à des applications pour une catégorie particulière de systèmes, à savoir les systèmes de transport.

2.1 Modélisation des systèmes à événements discrets dans l'algèbre des dioïdes

2.1.1 Système à événements discrets

Définition 2.1 (Système à événements discrets)
Un système à événements discrets est un système dont l'espace d'état est discret. Les transitions d'état sont provoquées par l'occurrence d'événements asynchrones. L'ensemble des événements est également discret. Un événement est validé si l'ensemble des conditions nécessaires à son occurrence sont vérifiées. Un événement validé peut survenir et son occurrence est instantanée. Pour prendre en compte le temps, des temporisations peuvent être associées aux événements : une temporisation est un laps de temps qui doit s'être écoulé avant qu'un événement validé puisse survenir.

Au moins deux niveaux d'abstraction peuvent être considérés pour un système à événements discrets :
- on peut s'intéresser exclusivement à l'ordre d'occurrence des événements, c'est-à-dire au *comportement logique* du système ;
- en plus de l'ordre, on peut étudier les instants d'occurrence des événements, c'est-à-dire le *comportement temporisé* du système [1].

Plusieurs objets mathématiques ont été proposés pour modéliser les systèmes à événements discrets. Les réseaux de Petri constituent un de ces formalismes dont on précise ici les notations, ainsi que sa capacité à capturer les deux niveaux d'abstraction cités (la littérature sur les réseaux de Petri est abondante ; citons les synthèses [Murata, 1989, David and Alla, 1989]).

Définition 2.2 (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 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), $M : \mathcal{P} \to \mathbb{N}$ définit le marquage initial des places (représenté par des jetons dans les places)* [2].

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.

Lorsque l'on étudie un système à événements discrets *via* un réseau de Petri :
- l'*état* est traduit par le *marquage du réseau* ;
- les *événements* sont représentés par les *transitions* (un franchissement d'une transition traduit une occurrence de l'événement associé).

La définition précédente des réseaux de Petri permet seulement d'appréhender le comportement logique des systèmes à événements discrets. Pour étudier leur comportement temporisé, la

1. Dans la littérature, la nuance est parfois soulignée en parlant alors de *système dynamique à événements discrets*. Dans ce manuscrit, on préfère utiliser de façon générique la dénomination *système à événements discrets*.

2. On se restreint aux réseaux de Petri qui sont *ordinaires* (poids unitaire pour chaque arc), et pour lesquels à chaque transition est associé un événement différent (dans la littérature anglaise, on parle de *free-labelled Petri nets*).

définition a été étendue en associant des temporisations aux transitions et/ou places. On parle alors de réseaux de Petri temporisés [3] [Ramchandani, 1973]. Pour les réseaux de Petri temporisés au niveau des transitions (parfois qualifiés de *T-temporisés*)), une durée finie de franchissement τ est associée avec chaque transition : τ est le temps minimal qui doit s'écouler avant qu'une transition validée puisse être franchie [4].

L'ensemble des marquages atteignables d'un réseau de Petri $\mathcal{G}$ à partir du marquage initial M est noté $\mathcal{R}(\mathcal{G}, M)$. Si le réseau est *borné* [5], celui-ci peut être représenté par le *graphe des marquages*, à savoir, un graphe orienté dont les noeuds sont les marquages et dont les arcs sont étiquetés par les transitions franchies lors de l'évolution du marquage [6].

Pour fixer les idées quant à la dimension générique des classes de systèmes considérés, nous distinguerons trois types d'événements qui peuvent être mis en jeu dans un système à événements discrets [7] :

Décalage
: L'occurrence d'un événement a peut causer la validation d'un événement b avec un décalage dans le temps et/ou dans le décompte des occurrences. D'une part, l'occurrence de l'événement a à un instant t peut impliquer que l'événement b est validé à partir de t et peut survenir à partir de $t + \tau$ (τ correspond à la temporisation associée à l'occurrence de b). D'autre part, la k-ième occurrence de l'événement a peut valider l'événement b pour la $k + n$-ième fois.

Synchronisation
: Deux événements a et b se trouvent en *synchronisation* si la conjonction de leurs occurrences est nécessaire à l'évolution du système (la validation d'un tiers événement) [8], ou si l'occurrence d'un tiers événement entraîne les validations simultanées, puis les occurrences, de a et b.

Concurrence
: Deux événements a et b se trouvent en *concurrence* si, l'occurrence d'un de ces événements provoque indistinctement la validation d'un tiers événement [9]. On parle également de concurrence et plus précisément de *conflit* lorsqu'un même événement provoque la validation de deux événements a et b et que l'occurrence de l'un exclut l'occurrence de l'autre.

Sur la figure 2.1, on a schématisé les configurations au sein d'un réseau de Petri modélisant chacun de ces phénomènes.

2.1.2 Modélisation du comportement temporisé dans l'algèbre des dioïdes

Pour l'étude du comportement temporisé des systèmes à événements discrets, deux types de variables sont communément utilisés [Cohen et al., 1986].

3. Ou de réseaux de Petri temporels si une définition des temporisations plus générale est considérée [Berthomieu and Diaz, 1991].

4. Dans le cas P-temporisé, la temporisation associée à une place correspondant à un temps de séjour minimal avant qu'un jeton puisse contribuer au franchissement de la (des) transition(s) de sortie.

5. C'est-à-dire, si le nombre de jeton(s) contenu(s) dans chaque place est fini.

6. Pour tout M, $M' \in \mathcal{R}(\mathcal{G}, M)$, il existe un arc de M à M' étiqueté par a si, et seulement si, a est validée par M et son franchissement transforme M en M'.

7. Ils apparaissent dans la littérature comme un jeu de phénomènes élémentaires qui, en les combinant, permettent de traduire un large spectre des *dynamiques* possibles au sein d'un système à événements discrets.

8. La validation du tiers événement est alors *synchronisée* avec l'occurrence des événements a et b, ce qui justifie la dénomination.

9. Les occurrences de a et b peuvent concurremment valider un tiers événement.

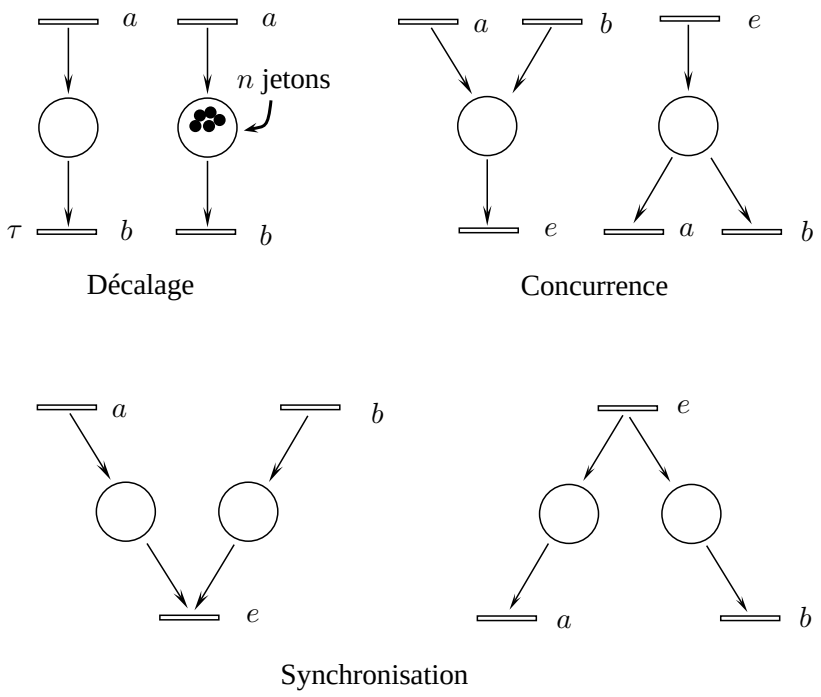

Figure 2.1 : Configurations au sein d'un réseau de Petri modélisant les phénomènes identifiés au sein d'un système à événements discrets.

Définition 2.3 (Dateur, Compteur)

— *On appelle dateur une fonction $d(k)$, $k \in \mathbb{Z}$, qui désigne la date de la $k+1$-ième occurrence de l'événement référencé par la lettre d.*
— *On appelle compteur une fonction du temps $c(t)$, $t \in \mathbb{Z}$, qui désigne le nombre cumulé d'occurrences de l'événement "c" ayant eu lieu jusqu'à la date t.*

Avec ces variables associées aux événements, on peut examiner les relations qu'induisent les phénomènes pouvant survenir au sein d'un système (voir notamment la figure 2.1 pour une description sous la forme d'un réseau de Petri) :

Décalage

Le fait que l'occurrence de l'événement a à un instant t implique que l'événement b est validé à partir de t et peut survenir à partir de $t+\tau$ se traduit par la relation suivante entre les compteurs associés aux événements :

$$b(t+\tau) \leq a(t)\,, \tag{2.1}$$

ou encore, si des dateurs sont plutôt considérés :

$$b(k) \geq a(k) + \tau\,. \tag{2.2}$$

Le fait que la k-ième occurrence de l'événement a valide l'événement b pour la $k+n$-ième fois se traduit par la relation suivante entre les compteurs associés aux événements :

$$b(t) \leq a(t) + n\,, \tag{2.3}$$

ou encore, si des dateurs sont plutôt considérés :

$$b(k+n) \geq a(k) \,. \tag{2.4}$$

Synchronisation

Si un événement e est validé une fois que deux événements a et b ont eu lieu, on a alors la relation suivante entre les compteurs associés aux événements :

$$e(t) \leq \min(a(t), b(t)) \,, \tag{2.5}$$

ou encore, si des dateurs sont plutôt considérés :

$$e(k) \geq \max(a(k), b(k)) \,. \tag{2.6}$$

Si l'occurrence d'un événement e induit que deux événements a et b sont validés simultanément, les compteurs vérifient alors :

$$a(t) \leq e(t) \text{ et } b(t) \leq e(t) \,, \tag{2.7}$$

alors que si des dateurs sont associés aux événements, ceux-ci satisfont :

$$a(k) \geq e(k) \text{ et } b(k) \geq e(k) \,. \tag{2.8}$$

Concurrence

Si l'occurrence de l'événement a ou de l'événement b provoque indistinctement la validation de l'événement e, on a alors la relation suivante entre les compteurs associés aux événements :

$$e(t) \leq a(t) + b(t) \,, \tag{2.9}$$

mais aucune relation simple peut être écrite entre les dateurs.

Si l'occurrence d'un événement e provoque la validation de a et b, et que l'occurrence de l'un exclut l'occurrence de l'autre, on a alors la relation suivante entre les compteurs :

$$a(t) + b(t) \leq e(t) \,, \tag{2.10}$$

et, à nouveau, aucune relation simple peut être écrite entre les dateurs.

La définition 2.1 précise qu'un événement validé et dont la temporisation s'est écoulée peut survenir. Il faut souligner que cela n'indique pas précisément l'instant d'occurrence de l'événement. Cet "indéterminisme" conduit aux inégalités mises en jeu dans (2.1)-(2.10). Si on fait l'hypothèse d'un *fonctionnement au plus tôt*, alors ces relations deviennent des égalités.

Définition 2.4 (Fonctionnement au plus tôt) *On dit qu'un système à événements discrets a un fonctionnement au plus tôt si chaque événement survient dès qu'il est validé et que sa temporisation est écoulée.*

Seuls les opérateurs max, min et + sont mis en jeu dans les relations (2.1)-(2.10), et c'est ce qui a conduit des chercheurs à initier une étude en exprimant la dynamique des systèmes à événements discrets dans les algèbres dites $(\max, +)$ et $(\min, +)$ (par des relations alors linéaires). Nous rappelons plusieurs définitions nécessaires pour introduire clairement ces structures algébriques.

Définition 2.5 (Monoïde) *Un monoïde $(\mathcal{M}, \oplus)$ est un ensemble $\mathcal{M}$ muni d'une loi de composition interne, notée $\oplus$, associative et possédant un élément neutre noté ε ($\forall m \in \mathcal{M}$, $m \oplus \varepsilon = \varepsilon \oplus m = m$). Si la loi $\oplus$ est commutative, le monoïde est dit commutatif.*

Définition 2.6 (Semi-anneau, Dioïde) *On appelle semi-anneau un ensemble $\mathcal{D}$ muni de deux lois internes $\oplus$ et $\otimes$ tel que :*

- *$(D, \oplus)$ est un monoïde commutatif dont l'élément neutre ε est appelé élément nul.*
- *$(D, \otimes)$ est un monoïde. Son élément neutre est appelé unité et est noté e.*
- *La loi multiplicative $\otimes$ est distributive à droite et à gauche par rapport à la loi additive $\oplus$.*
- *L'élément nul ε est absorbant pour la loi $\otimes$ ($\forall a \in \mathcal{D}$, $a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$).*

En outre, si la loi additive $\oplus$ est idempotente ($\forall a \in \mathcal{D}$, $a \oplus a = a$), alors $(\mathcal{D}, \oplus, \otimes)$ est qualifié de semi-anneau idempotent ou dioïde.

On dit qu'un dioïde est commutatif dès que la loi multiplicative est commutative. Lorsqu'aucune confusion ne sera possible, on omettra le signe multiplicatif '$\otimes$'.

Exemple 2.7 (Algèbres (max,+) et (min,+)) Les dioïdes $\overline{\mathbb{R}}_{\max} = (\mathbb{R} \cup \{-\infty, +\infty\}, \max, +)$ et $\overline{\mathbb{Z}}_{\max} = (\mathbb{Z} \cup \{-\infty, +\infty\}, \max, +)$, pour lesquels $\varepsilon = -\infty$ et $e = 0$, sont souvent appelés "algèbre $(\max, +)$".
Les "algèbres $(\min, +)$" correspondent à $\overline{\mathbb{R}}_{\min} = (\mathbb{R} \cup \{-\infty, +\infty\}, \min, +)$ et $\overline{\mathbb{Z}}_{\min} = (\mathbb{Z} \cup \{-\infty, +\infty\}, \min, +)$ pour lesquels $\varepsilon = +\infty$ et $e = 0$. ◇

Les relations mettant en jeu des compteurs (2.1), (2.3), (2.5), (2.7), (2.9), (2.10) peuvent se réécrire dans l'algèbre $(\min, +)$. Les compteurs constituent les variables d'état du système étudié. Plus formellement, l'ensemble des compteurs peut être muni d'une structure de moduloïde (sorte d'espace vectoriel bâti sur le monoïde $(\mathbb{R}, \min)$). Considérant un système comme une application définie sur cet ensemble de signaux, une théorie des systèmes spécifique a été développée [10] [Baccelli et al., 1992, Gaubert, 1992].

2.1.3 Systèmes linéaires dans l'algèbre des dioïdes

Les phénomènes de synchronisation et de décalage peuvent être décrits par des équations linéaires dans l'algèbre des dioïdes (en considérant un fonctionnement au plus tôt, *cf.* définition 2.4). En particulier si on considère des compteurs et que l'on exprime les relations (2.1)-(2.3)-(2.5)-(2.7) dans l'algèbre $(\min, +)$, on obtient respectivement :

Décalage

$$b(t + \tau) = a(t) \,, \tag{2.11}$$

$$b(t) = n \otimes a(t) \,, \tag{2.12}$$

Synchronisation

$$e(t) = a(t) \oplus b(t) \,, \tag{2.13}$$

$$a(t) = e(t) \text{ et } b(t) = e(t) \,. \tag{2.14}$$

Il a été mis en évidence que la classe des systèmes mettant en jeu exclusivement ces phénomènes (synchronisation, décalage) coïncide avec celle des systèmes qui peuvent être représentés par des graphes d'événements [11] temporisés, ou encore avec la classe des systèmes linéaires stationnaires

10. On peut aussi raisonner à partir de dateurs dans l'algèbre $(\max, +)$.
11. Réseaux de Petri pour lesquels chaque place a exactement une transition en amont et une transition en aval.

dans les algèbres $(\min, +)$ ou $(\max, +)$. Cette classe a fait l'objet de nombreuses études dans la littérature aboutissant à des résultats d'analyse, d'évaluation de performance et/ou de commande des systèmes à événements discrets. Pour la seule problématique de commande [12] citons (sans viser l'exhaustivité) :

- la commande optimale en boucle ouverte [Cohen et al., 1989, Menguy et al., 2000],
- la commande à modèle interne [Boimond and Ferrier, 1996],
- la commande prédictive [De Schutter and van den Boom, 2001], [De Schutter and van den Boom, 2003],
- la synthèse de correcteurs de type retour de sortie et retour d'état [Cottenceau et al., 2001], [Maia et al., 2003],
- la synthèse de contrôleurs robustes [Lhommeau et al., 2004].

2.2 Des travaux visant à étendre les classes des systèmes étudiés dans l'algèbre des dioïdes

Les articles fondateurs de la théorie des systèmes dans les dioïdes se sont focalisés sur les systèmes linéaires stationnaires, c'est-à-dire la classe des systèmes mettant en jeu exclusivement des phénomènes de synchronisation et décalage, ou encore la classe de systèmes à événements discrets qui peuvent être représentés par des graphes d'événements temporisés. Depuis, de nombreux travaux ont contribué à mettre en évidence que le formalisme reposant sur les dioïdes peut contribuer à l'étude d'une classe plus large de systèmes à événements discrets. C'est une préoccupation qui a, depuis toujours et de façon constante, motivé une partie de mes travaux de recherche. Pour en témoigner, mentionnons

- que des travaux dans la continuité de ceux de ma thèse ont porté sur les systèmes linéaires mais non stationnaires sur les dioïdes. Dans ce cadre, des résultats de représentation, d'évaluation de performance et de commande ont pu être étendus à la classe de systèmes à événements discrets qui peuvent être représentés par des *graphes d'événements avec des temporisations variables* et des *graphes d'événements avec ajout/retrait dynamique de jetons* [Lahaye et al., 2004, Lahaye et al., 2008] ;
- qu'une collaboration sur certains papiers de Samir Hamaci a permis d'étudier des systèmes à événements discrets pouvant être représentés par des *graphes d'événements temporisés hybrides* [13] *avec multiplieurs* [14] [Hamaci et al., 2006b, Hamaci et al., 2006a], [Hamaci et al., 2007]. ;
- que durant la thèse de Michell Al Saba [T2], on s'est intéressé aux *modèles de types tas* qui admettent une description reposant sur les dioïdes et qui permettent d'étudier tout système à événements discret pouvant être représenté par un *réseau de Petri sauf* (soulignons que des phénomènes de concurrence peuvent alors être considérés). Par le biais de ces modèles, des systèmes flexibles de production manufacturière ont pu être étudiés et une commande de type juste-à-temps a été proposée [Alsaba et al., 2007, Alsaba et al., 2006b,

12. Pour laquelle les résultats sont des transpositions de ceux bien connus en théorie conventionnelle des systèmes.

13. Le caractère *hybride* provient du fait que dans certaines parties du réseau de Petri on considère que les places contiennent un "liquide" (quantifié par des valeurs réelles), au lieu d'un nombre discret de jetons, qui "s'écoule" au travers des transitions. Par ce biais, on modélise de façon approchée, mais avec une faible complexité, des systèmes qui conduiraient à un réseau de Petri (non hybride) contenant un nombre très important de jetons (ce qui induit un graphe des marquages, en l'occurrence un espace d'état, très étendu.)

14. Des poids sur les arcs du réseau de Petri permettent en particulier de modéliser simplement l'assemblage et le désassemblage de lots de produits.

Alsaba et al., 2006a, Alsaba et al., 2006c].
- que la collaboration [C1] (dans la continuité de ces derniers travaux) et la thèse [T4] encours portent sur les *automates* $(\max, +)$ (les modèles de type tas sont des automates $(\max, +)$ particuliers).

On se focalise dans la suite de cette section sur les travaux menés depuis quatre ans sur les automates $(\max, +)$. Plus exactement, on en fait une synthèse en essayant de faire ressortir la cohérence de la démarche scientifique. On commence par quelques brefs rappels sur les automates $(\max, +)$ dont la substance provient des articles fondateurs [Gaubert, 1995] et [Gaubert and Mairesse, 1999] [15]. Une démarche pour la modélisation des systèmes à événements discrets à l'aide d'automates $(\max, +)$ est ensuite proposée. Dans cette partie, on expose en particulier deux moyens pour traduire la composition synchrone d'automates $(\max, +)$: la première est à notre connaissance originale et n'a pas encore été publiée, la deuxième a fait l'objet de la communication [Komenda et al., 2009b]. Une approche pour la commande des automates $(\max, +)$ est ensuite présentée (on évitera d'exposer les aspects techniques que le lecteur trouvera dans la publication [Komenda et al., 2009c] jointe en annexe).

2.2.1 Automates $(\max, +)$

Définition 2.8. *Un automate (max,+) est un quintuplet $G = (Q, A, q_0, t, Q_m)$, où Q est l'ensemble fini d'états, A est l'ensemble fini d'événements (encore appelé alphabet), $t : Q \times A \times Q \to \overline{\mathbb{R}}_{\max}$ est la fonction de transition, q_0 (resp. Q_m) est l'état initial (resp. l'ensemble des états finaux).*

La fonction de transition associe à un état $q \in Q$, un événement $a \in A$ et un état $q' \in Q$, une valeur $t(q, a, q') \in \mathbb{R} \cup \{\pm\infty\}$ correspondant à la durée de transition a de q vers q' (avec la convention que $t(q, a, q') = \varepsilon(= -\infty)$ si l'événement a ne permet pas de passer de l'état q à l'état q'). Plus précisément, $t(q, a, q')$ est la temporisation associée à l'événement a lors de la transition depuis l'état q vers l'état q'.

Nous dirons que l'automate est *déterministe* si $\forall q$, $\forall a$, il existe au plus un état q' tel que $t(q, a, q') \neq \varepsilon$.

Pour une séquence d'événements, c'est-à-dire, de lettres $a \in A$, qui peuvent survenir, on parle de *mot*, et l'ensemble des mots est appelé *langage* [16].

L'ensemble des séquences possibles (mots reconnus) et les durées correspondantes sont codées à l'aide de *séries formelles* [17]. Plus précisément, le *comportement d'un automate* (max,+) $G =$

15. Les automates à multiplicités dans un semi-anneau idempotent ont été étudiés bien avant, notamment par Imre Simon ou encore Kosaburo Hashiguchi. Les articles cités ont été, à notre connaissance, les premiers à mettre en évidence l'intérêt des automates $(\max, +)$ pour l'étude des systèmes à événements discrets.

16. 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 $(Pwr(A^*), \cup, .)$. Le langage nul est noté $0 = \{\}$, le langage unité est noté $1 = \{\varepsilon\}$ où ε est le mot vide.

17. Le *dioïde des séries formelles* à variables dans A et à coefficients dans $\overline{\mathbb{R}}_{\max}$, noté $\overline{\mathbb{R}}_{\max}[\![A]\!]$, est muni de l'addition mot à mot et d'une multiplication sous forme de convolution. Aussi, pour $s = \oplus_{w \in A^*} s(w)w$ (on note $s(w)$ le coefficient de la série s pour le mot w) et $s' = \oplus_{w \in A^*} s'(w)w \in \overline{\mathbb{R}}_{\max}[\![A]\!]$, on a :

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

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

Par exemple, avec $A = \{a, b\}$, $s = 1a \oplus 2ab$ et $s' = 3ab$, on a $s \oplus s' = 1a \oplus 3ab$ et $s \otimes s' = 4aab \oplus 5abab$. Le langage $supp(s) = \{w \in A^* : s(w) \neq -\infty\}$ est appelé le *support de la série* s, on a $supp(s) = \{a, ab\}$

(Q, A, q_0, t, Q_m) est donné par la série formelle $l(G) \in \overline{\mathbb{R}}_{\max}[\![A]\!]$ suivante :

$$\forall w = a_1 \dots a_n \in A^*, \quad l(G)(w) = \max_{q_0,\dots,q_n \in Q} \alpha_{q_0} + \left[\sum_{i=1}^{n} t(q_{i-1}, a_i, q_i) \right] + \beta_{q_n}, \tag{2.15}$$

où q_n est un état final. Autrement dit, $l(G)(w)$ est le poids maximal de tous les chemins valués par w menant de l'état initial vers un état final. Soulignons que si l'automate est déterministe, alors il n'existe qu'un chemin reconnaissant chaque mot w et l'opérateur max n'intervient plus dans (2.15).

Remarque 2.9 La série formelle $l(G)$ est un dateur généralisé [Gaubert, 1995], où $l(G)(w)$ représente la date à laquelle la séquence d'événements (tâches) w a été achevée (avec la convention $l(G)(w) = \varepsilon$ si w n'a pas lieu). La spécialisation aux séries booléennes à coefficients dans $\{\varepsilon, e\}$ ramène à la notion classique de langage dans la théorie de Ramadge et Wonham [Ramadge and Wonham, 1989], c'est-à-dire, $l(G)(w) = e$ si w est un comportement admissible de l'automate. $\diamond$

Un automate (max,+) est aussi souvent défini par un triplet, appelé *représentation linéaire*, (α, μ, β) où $\alpha \in \overline{\mathbb{R}}_{\max}^{1\times|Q|}$ (resp. $\beta \in \overline{\mathbb{R}}_{\max}^{|Q|\times 1}$) est tel que $\alpha_q \neq \varepsilon$ (resp. $\beta_q \neq \varepsilon$) si q est l'état initial (resp. un état final), $\alpha_q = \varepsilon$ (resp. $\beta_q = \varepsilon$) sinon, et μ est le *morphisme*

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

De la même manière que les graphes d'événements temporisés admettent une description par des équations linéaires implicites dans le dioïde des séries formelles $\mathbb{Z}_{\max}(\gamma)$, [Baccelli et al., 1992, §5.3], tout automate (max,+) peut être décrit dans le dioïde $\overline{\mathbb{R}}_{\max}[\![A]\!]$ par le système d'équations :

$$\left\{ \begin{array}{lcl} x & = & x\mu \oplus \alpha \\ y & = & x\beta \end{array} \right. ,$$

où $\mu = \bigoplus_{a\in A} \mu(a)a \in \overline{\mathbb{R}}_{\max}[\![A]\!]^{|Q|\times|Q|}$ est la matrice de morphisme. Rappelons que sa plus petite solution (traduisant le fonctionnement au plus tôt de l'automate) est donnée par $y = l(G) = \alpha \otimes \mu^* \otimes \beta$.

Exemple 2.10 La figure 2.2 est un exemple de représentation graphique qui peut être associée à chaque automate $(\max, +)$: les sommets correspondent aux états, des arcs pondérés avec les événements et leurs temporisations représentent les transitions d'états, des arcs d'entrée (resp. de sortie) signifient les états initiaux (resp. finaux). Pour cet exemple,

- l'ensemble des événements est $A = \{a, b, c\}$,
- l'ensemble des états est $Q = \{I, II\}$,
- l'état initial est $q_0 = \{I\}$,
- Q_m est un singleton, $Q_m = \{II\}$,
- les temporisations associées aux événements a, b et c valent respectivement 2, 1 et 3 unités de temps.

Les séquences d'événements possibles sont : a, ac, ab, acc, acb, aba, $accc$, $accb$, $acba$, $abac$, $abab$, ... La représentation linéaire de G est donnée par

$$\alpha = \begin{pmatrix} e & \varepsilon \end{pmatrix}, \ \mu(a) = \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & \varepsilon \end{pmatrix}, \ \mu(b) = \begin{pmatrix} \varepsilon & \varepsilon \\ 1 & \varepsilon \end{pmatrix},$$

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

Les matrices μ sont construites en indexant les lignes et colonnes avec les éléments de Q, ici $\{I, II\}$. Par exemple,

$$\mu(a) = \begin{array}{cc} & \begin{array}{cc} I & II \end{array} \\ \begin{array}{c} I \\ II \end{array} & \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & \varepsilon \end{pmatrix} \end{array}$$

et $[\mu(a)]_{I,II} = 2$ s'interprète en relevant qu'il existe une transition de l'état I vers l'état II selon a avec une temporisation de 2 unités de temps. Le comportement de G peut être déduit en calculant

$$\begin{aligned} y = l(G) = \alpha \otimes \mu^* \otimes \beta &= \alpha \otimes (\mu(a)a \oplus \mu(b)b \oplus \mu(c)c)^* \otimes \beta, \\ &= \alpha \otimes \begin{pmatrix} \varepsilon & 2a \\ 1b & 3c \end{pmatrix}^* \otimes \beta. \end{aligned}$$

On obtient la série de $\overline{\mathbb{R}}_{\max}[\![A]\!]$ suivante :

$$y = 2a(3c \oplus 3ba)^*.$$

Par exemple, $y(aba) = 5$ signifie que la séquence aba sera achevée à la date 5 (en considérant que le système fonctionne au plus tôt à partir de l'instant 0).

◇

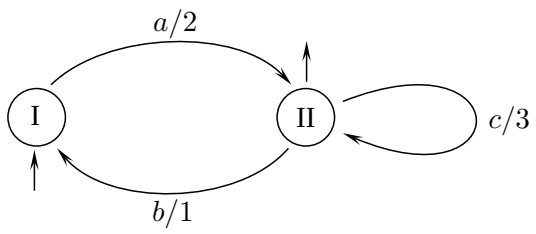

Figure 2.2 : Un automate (max,+).

2.2.2 Modélisation des systèmes à événements discrets à l'aide d'automates (max, +)

Les automates (max, +) sont particulièrement appropriés à la modélisation des phénomènes de décalage et de concurrence. Pour s'en convaincre, on peut considérer l'exemple représenté sur la figure 2.2 (avec un fonctionnement au plus tôt) :

Décalage

L'occurrence de l'événement b provoque la validation de a qui survient une fois sa temporisation écoulée (2 unités de temps).

Concurrence

L'occurrence de l'événement a provoque un *conflit* : b et c sont alors simultanément validés mais un seul de ces événements peut survenir (l'occurrence de l'un exclut l'occurrence de l'autre).

Stéphane Gaubert et Jean Mairesse ont montré que les dates de franchissement des transitions de tout réseau de Petri sauf[18] peuvent être obtenues à l'aide d'un automate (max, +). Plus

18. Chaque place peut contenir au plus un jeton.

formellement, pour tout réseau de Petri sauf $\mathcal{G}$, il existe un automate $(\max, +)$ G tel que si w est une séquence de franchissements de transitions possibles, alors $l(G)(w)$ (*cf.* remarque 2.9) correspond à la date à laquelle s'achève la séquence de franchissements [Gaubert and Mairesse, 1999]. Aucune hypothèse n'est formulée quant à la structure du réseau de Petri, et des phénomènes de synchronisation peuvent donc être *a priori* pris en compte. Considérons par exemple le réseau de Petri représenté sur la figure 2.3. Son évolution (au sens ici des dates auxquelles s'achèvent les séquences de franchissements possibles) peut être traduit par le comportement de l'automate représenté sur la figure 2.4. Précisons :

- comment la transition à franchir est choisie lors d'un conflit : tous les choix logiques possibles sont considérés (en particulier, la décision n'est pas basée sur des considérations temporelles [19]) ;
- qu'une définition plus générale que la déf. 2.8 est considérée pour les automates $(\max, +)$ car ceux-ci peuvent comporter plusieurs états initiaux.

L'automate de la figure 2.4 est non déterministe : depuis certains états plusieurs transitions sont possibles sur l'occurrence d'un même événement (depuis l'état I, l'occurrence de a conduit soit à l'état II, soit à l'état IV) et il y a deux états initiaux (I et III). C'est le "prix" à payer pour modéliser les phénomènes de synchronisation (ce qui, en d'autres termes, revient à faire effectivement intervenir l'opérateur max dans (2.15)). L'obtention de cet automate n'est pas très aisée. Plus généralement, la tâche de "l'ingénieur", qui cherche à modéliser à l'aide d'un automate $(\max, +)$ un système comportant des phénomènes de synchronisation, nous semble plutôt difficile car elle requiert une bonne expertise des automates $(max, +)$. De plus, il y a plusieurs solutions possibles (plusieurs automates $(\max, +)$ peuvent permettre de calculer les dates de franchissements des transitions d'un même réseau de Petri) sans qu'à notre connaissance, il ne soit possible d'identifier un représentant minimal. Le constat précédent nous a conduit à ébaucher une approche de modélisation modulaire. Celle-ci consiste à décomposer le système à événements discrets en sous-systèmes mettant en jeu exclusivement des phénomènes de concurrence et décalage. Les interactions entre ces sous-systèmes sont prises en compte au travers d'événements communs survenant simultanément en leur sein (donnant lieu aux synchronisations). Cette démarche nous semble naturelle et trouve des analogies fortes dans la littérature des réseaux de Petri : elle revient à modéliser des sous-systèmes à l'aide de graphes d'états et à composer ceux-ci en fusionnant des transitions. Cette décomposition est notamment utilisée dans [Gaubert and Mairesse, 1999] pour réduire la taille du *modèle de type tas* représentant un réseau de Petri sauf, duquel on peut déduire un *automate de type tas* et finalement un automate $(\max, +)$. L'approche proposée peut donc être vue comme une alternative (pour la modélisation) grâce à laquelle on "fait l'économie" des étapes intermédiaires mettant en jeu les modèles et automates de types tas.

2.2.2.1 Automate déterministe comme brique élémentaire de modélisation

Il est pris ici comme point de vue que la "brique élémentaire" de modélisation [20] est l'automate $(\max, +)$ déterministe. Comme nous l'avons déjà remarqué, dans un automate déterministe, la durée d'une séquence est simplement obtenue comme la somme des durées des événements qui la composent (l'opérateur max n'intervient pas dans (2.15)). Cela signifie notamment que les phénomènes de synchronisation ne peuvent pas être pris en compte. Ceci conduit à une proposition assez évidente, mais qui a pour intérêt de fixer les idées quant à la classe des systèmes qui peut être appréhendée sous la forme d'automate $(\max, +)$ déterministe.

19. A l'opposé, la *politique de la course* est souvent considérée dans la littérature.
20. Dans le sens où le modèle d'un système va être obtenu à partir d'un assemblage de telles briques élémentaires.

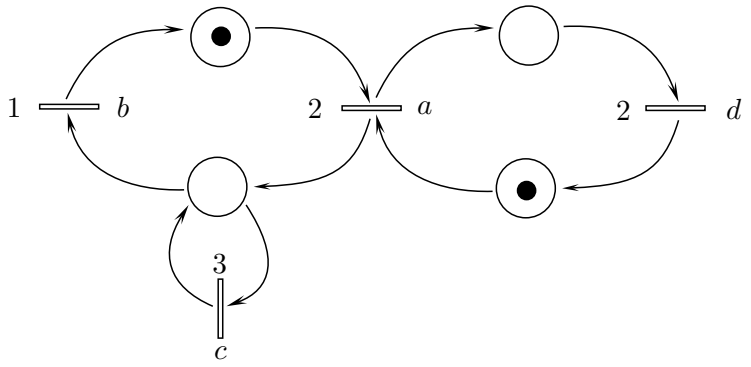

Figure 2.3 : Un réseau de Petri 1-sauf

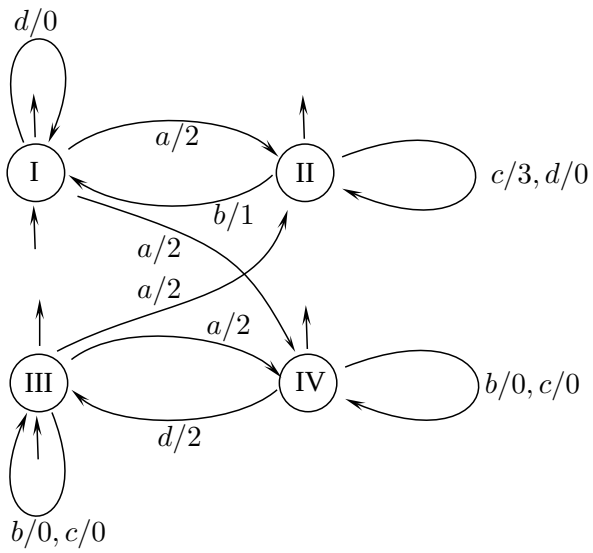

Figure 2.4 : Automate (max,+) dont le comportement reconnaît l'évolution du réseau de Petri représenté sur la figure 2.3

Proposition 2.11. *La classe des systèmes à événements discrets qui peuvent être modélisés par un automate* $(\max, +)$ *déterministe coïncide avec celle des systèmes qui peuvent être représentés par un graphe d'état*[21] *sauf.*

Preuve : Commençons par noter que tout graphe d'état temporisé sauf $\mathcal{G} = (\mathcal{P}, \mathcal{T}, \mathcal{F}, M)$ peut être transformé en un automate $(\max, +)$ déterministe $G = (Q, A, q_0, t, Q_m)$ ayant le même comportement (dans le sens où les franchissements des transitions dans $\mathcal{G}$ ont lieu aux mêmes instants que les transitions d'état correspondantes dans G). La construction de G consiste à prendre pour

- ensemble des états Q l'ensemble des marquages atteignables $\mathcal{R}(\mathcal{G}, M)$ (qui est fini puisque le graphe d'état est sauf) ;
- ensemble des événements A l'ensemble fini $\mathcal{T}$ des transitions (à chaque transition est associé un événement différent) ;

21. Un réseau de Petri au sein duquel chaque transition a exactement une place en amont et une place en aval.

- état initial q_0 le marquage initial M ;
- fonction de transition la relation définissant les arcs de $\mathcal{R}(\mathcal{G}, M)$ pondérés par les temporisations et les labels des transitions ($t(M, a, M') = \tau$ si a est validée avec le marquage M, si son franchissement transforme le marquage M en M', et si sa temporisation vaut τ unités de temps).
- ensemble des états finaux Q_m l'ensemble des marquages atteignables $\mathcal{R}(\mathcal{G}, M)$ (tous les états sont considérés finaux).

L'automate ainsi obtenu est déterministe (il ne le serait éventuellement pas si un même événement avait été associé à plusieurs transitions distinctes). Cette construction peut être réalisée pour tout réseau de Petri borné mais les transitions d'état dans l'automate coïncident avec les franchissements des transitions seulement s'il n'y a pas de synchronisation. En effet, comme il est déterministe, le comportement de l'automate est donné par l'équation (2.15) sans que l'opérateur max n'intervienne, c'est-à-dire,

$$l(G)(w) = \alpha(q_1) + \sum_{i=1}^{n-1} t(q_i, a_i, q_{i+1}) + \beta(q_n) \,.$$

En d'autre termes, les événements (franchissements dans le réseau de Petri) dans w doivent survenir de façon purement séquentielle (les uns à la suite des autres sans simultanéité). Dans un graphe d'état sauf, une seule transition peut être franchie à la fois, et le comportement de l'automate $(\max, +)$ construit traduit donc bien son évolution (les dates auxquelles surviennent les franchissements).

Dans le sens inverse, il est possible d'associer à tout automate $(\max, +)$ déterministe $G = (Q, A, q_0, t, Q_m)$ un graphe d'état sauf $\mathcal{G} = (\mathcal{P}, \mathcal{T}, \mathcal{F}, M)$ ayant le même comportement. On procède comme suit.

1. À chaque état dans Q on associe une unique place dans P, c'est-à-dire que l'on pose $P = Q$. Cela précise en outre le marquage initial M de $\mathcal{G}$: il s'agit du vecteur ligne $\begin{pmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{pmatrix}$ dans lequel le seul coefficient non nul est pour la place dans P qui correspond à q_0 ;
2. À chaque triplet (q, a, q') dans G tel que $t(q, a, q') \neq \varepsilon$, on associe une transition dans $t \in \mathcal{T}$. Il y a donc autant de transitions dans $\mathcal{G}$ qu'il y a d'arcs dans la représentation graphique de G. On définit alors
   - le label et la temporisation de la transition par l'événement $a \in A$ et $t(q, a, q')$;
   - deux arcs dans $\mathcal{F}$ pour (q, t) et (t, q').

   Cette application qui a une transition d'état de G associe une transition dans $\mathcal{G}$ est illustrée sur la figure 2.5.

Si l'automate est déterministe, alors le réseau de Petri ainsi construit est tel que chaque transition a exactement une place d'entrée et une place de sortie [22], ce qui correspond à un graphe d'état. Comme de surcroît une seule place contient initialement un jeton (car il n'y a qu'un état initial q_0 dans G) le graphe d'état est sauf.

□

2.2.2.2 Composition d'automates $(\max, +)$ déterministes

On aborde ici la composition de plusieurs automates $(\max, +)$ déterministes (correspondant aux briques de modélisation). Afin d'alléger la présentation, on se limite dans la suite à deux

22. Notez en revanche que dans ce réseau de Petri, plusieurs transitions peuvent correspondre à un même événement et donc avoir le même label.

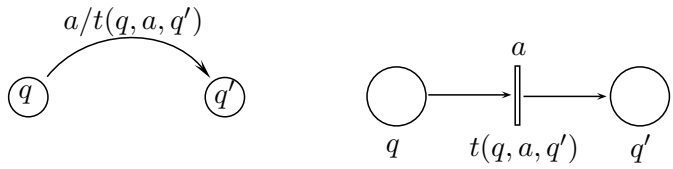

Figure 2.5 : Illustration de l'application qui a une transition d'état d'un automate fait correspondre une transition dans un réseau de Petri.

automates $(\max,+)$ $G_1 = (Q_1, A_1, \alpha_1, t_1, \beta_1)$ et $G_2 = (Q_2, A_2, \alpha_2, t_2, \beta_2)$. On parle d'*événement commun* pour un événement survenant au sein des deux automates $(\max,+)$ (dans $A_1 \cap A_2$) et d'*événement privé* dans le cas contraire. On précise dans la définition suivante le type de composition qui nous intéresse.

Définition 2.12 (Composition synchrone d'automates $(\max,+)$ **déterministes)** *La composition (ou produit) synchrone de G_1 et G_2 désigne un système dont l'évolution peut être décrite à partir de celles de G_1 et G_2 tel que*
- *l'occurrence d'un événement commun est simultanée au sein de G_1 et G_2 ;*
- *des séquences d'événements privés peuvent survenir de façon asynchrone au sein de G_1 et G_2 entre deux occurrences d'un événement commun.*

En termes de réseau de Petri, on considère des sous-systèmes représentés par des graphes d'état saufs (voir la preuve de la proposition 2.11), et on s'intéresse au système représenté par le réseau résultant de la fusion des transitions correspondant aux événements communs.
Pour représenter la composition synchrone, on propose dans ce qui suit deux approches. On montre en effet que la composition synchrone de G_1 et G_2 peut être traduite
- par un automate $(\max,+)$ non déterministe défini sur l'alphabet $A = A_1 \cup A_2$,
- et par un automate $(\max,+)$ déterministe défini sur un alphabet étendu noté $\mathcal{A}$.

Produit synchrone résultant en un automate non déterministe
On note
- $(\alpha_1, \mu_1, \beta_1)$ et $(\alpha_2, \mu_2, \beta_2)$ les représentations linéaires de G_1 et G_2,
- $A \triangleq A_1 \cup A_2$,
- $A_{com} \triangleq A_1 \cap A_2$.

Pour simplifier, on considère qu'une unique valeur de temporisation τ est associée à chaque événement commun $a \in A_{com}$ aux deux automates.

On définit ici un nouvel automate $(\max,+)$ noté $\mathfrak{G}$ sur l'alphabet A pour traduire la composition synchrone de G_1 et G_2.

Définition 2.13. *On définit la représentation linéaire (α, μ, β) de l'automate $(\max,+)$ $\mathfrak{G}$ défini sur l'alphabet A par :*

$$\alpha = \begin{pmatrix} \alpha_1 & \alpha_2 \end{pmatrix} ,$$

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} ,$$

$$\mu(a) = \left(\begin{array}{c|c} \mu_{11} & \mu_{12} \\ \hline \mu_{21} & \mu_{22} \end{array}\right) ,$$

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

– *pour $i = j$:*

$$\mu_{ii} = \begin{cases} \mu_i(a), & \text{si } a \in A_i \setminus A_{com}; \\ I_{|Q_i|}, & \text{sinon } (I_{|Q|} \text{ désigne la matrice identité dans } \overline{\mathbb{R}}_{\max} \text{ de} \\ & \text{dimension égale au nombre d'éléments d'un ensemble } Q). \end{cases}$$

– *pour $i \neq j$:*

$$[\mu_{ij}]_{kl} = \begin{cases} \tau, & \text{si } a \in A_{com} \text{ et } \exists q'_k \in Q_i, q'_l \in Q_j \\ & \text{tels que } t_i(q_k, a, q'_k) \neq \varepsilon \text{ et } t_j(q'_l, a, q_l) \neq \varepsilon; \\ \varepsilon, & \text{sinon.} \end{cases}$$

Assertion 1. *Pour tout mot $w \in A^*$ tel que $l(G_1)(P_1 w) \neq \varepsilon$ (P_1 désigne la projection sur le langage défini à partir de A_1* [23]*) et $l(G_2)(P_2 w) \neq \varepsilon$, la valeur du comportement de $\mathfrak{G}$*

$$l(\mathfrak{G})(w) = \alpha\mu(w)\beta$$

coïncide avec la date d'achèvement de la séquence w au sein de la composition synchrone de G_1 et G_2.

On se propose d'illustrer et d'interpréter la définition 2.13 et l'assertion 1 au travers de l'exemple suivant.

Exemple 2.14 Deux automates G_1 et G_2 sont représentés sur la partie gauche de la figure 2.6 (G_1 est identique à l'automate G considéré dans l'exemple 2.10). Leurs représentations linéaires sont

$$\alpha_1 = \begin{pmatrix} e & \varepsilon \end{pmatrix}, \ \mu_1(a) = \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & \varepsilon \end{pmatrix}, \ \mu_1(b) = \begin{pmatrix} \varepsilon & \varepsilon \\ 1 & \varepsilon \end{pmatrix},$$

$$\mu_1(c) = \begin{pmatrix} \varepsilon & \varepsilon \\ \varepsilon & 3 \end{pmatrix}, \ \beta_1 = \begin{pmatrix} \varepsilon \\ e \end{pmatrix},$$

$$\alpha_2 = \begin{pmatrix} e & \varepsilon \end{pmatrix}, \ \mu_2(a) = \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & \varepsilon \end{pmatrix}, \ \mu_2(d) = \begin{pmatrix} \varepsilon & \varepsilon \\ 2 & \varepsilon \end{pmatrix}, \ \beta_2 = \begin{pmatrix} \varepsilon \\ e \end{pmatrix}.$$

La composition synchrone est définie sur l'alphabet $A = A_1 \cup A_2 = \{a, b, c, d\}$ et l'ensemble des événements communs est $A_{com} = \{a\}$. Conformément à la définition 2.13, les matrices μ pour les événements privés sont (pour alléger les notations, on remplace dans les matrices suivantes les 'ε' par des '$\cdot$') :

$$\mu(b) = \left(\begin{array}{c|c} \mu_1(b) & \cdot \\ \hline \cdot & I_{|Q_2|} \end{array}\right) = \left(\begin{array}{cc|cc} \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & e & \cdot \\ \cdot & \cdot & \cdot & e \end{array}\right),$$

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

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

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

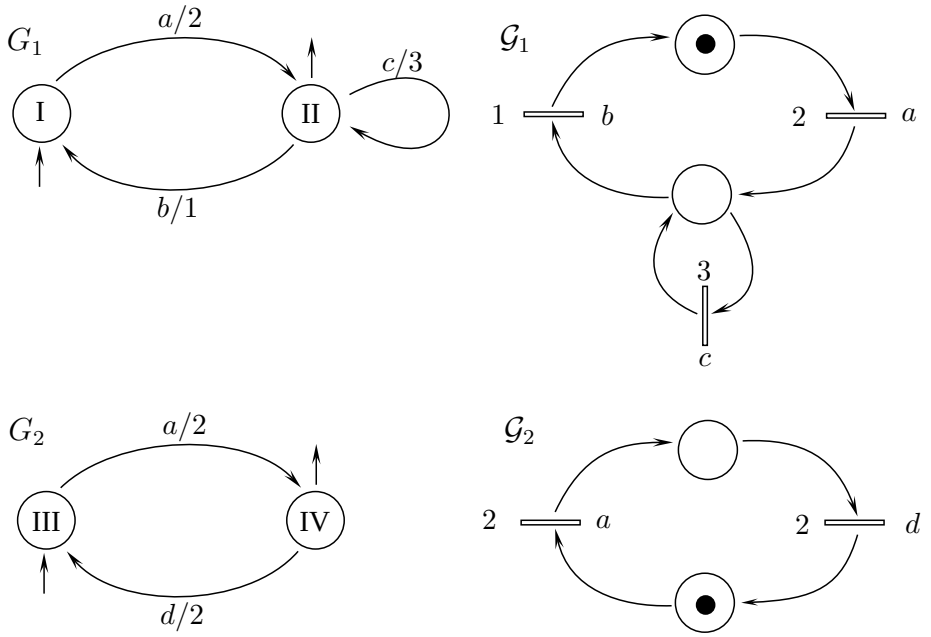

Figure 2.6 : Deux automates $(\max, +)$ et leurs graphes d'état équivalents.

$$\mu(c) = \left(\begin{array}{cc|cc} \cdot & \cdot & \cdot & \cdot \\ \cdot & 3 & \cdot & \cdot \\ \hline \cdot & \cdot & e & \cdot \\ \cdot & \cdot & \cdot & e \end{array}\right), \mu(d) = \left(\begin{array}{cc|cc} e & \cdot & \cdot & \cdot \\ \cdot & e & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 2 & \cdot \end{array}\right).$$

Pour interpréter la construction de ces matrices, rappelons que les lignes et colonnes sont indexées sur l'ensemble des états, ici $\{I, II, III, IV\}$. Par exemple,

$$\mu(b) = \begin{array}{c} \\ I \\ II \\ III \\ IV \end{array} \begin{array}{c} \begin{array}{cccc} I & II & III & IV \end{array} \\ \left(\begin{array}{cc|cc} \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & e & \cdot \\ \cdot & \cdot & \cdot & e \end{array}\right) \end{array}.$$

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

$$\mu(a) = \left(\begin{array}{cc|cc} \cdot & 2 & \cdot & 2 \\ \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & 2 & \cdot & 2 \\ \cdot & \cdot & \cdot & \cdot \end{array}\right).$$

Rappelons que dans la composition synchrone de G_1 et G_2 (*cf.* définition 2.12), on souhaite que l'occurrence d'un événement commun soit simultanée au sein de ces automates. La construction de la matrice $\mu(a)$ s'interprète alors 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.

  La définition 2.13 indique d'autre part que

$$\alpha = \left(\begin{array}{c|c} \alpha_1 & \alpha_2 \end{array} \right) = \left(\begin{array}{cc|cc} e & \cdot & e & \cdot \end{array} \right), \beta = \left(\begin{array}{c} \beta_1 \\ \hline \beta_2 \end{array} \right) = \left(\begin{array}{c} \cdot \\ e \\ \hline \cdot \\ e \end{array} \right).$$

La séquence $w = accbda$ sur A est telle que $l(G_1)(P_1 w) = l(G_1)(accba) \neq \varepsilon$ et $l(G_2)(P_2 w) = l(G_1)(ada) \neq \varepsilon$. Conformément à l'assertion 1, le calcul de $l(\mathfrak{G})(w) = \alpha\mu(w)\beta$ permet d'établir la date d'achèvement de la séquence dans la composition synchrone de G_1 et G_2. En effet, on a

$$\begin{aligned} l(\mathfrak{G})(accbda) &= \alpha\mu(a)\mu(c)\mu(c)\mu(b)\mu(d)\mu(a)\beta, \\ &= \left(\begin{array}{cc|cc} e & \cdot & e & \cdot \end{array} \right) \left(\begin{array}{cc|cc} \cdot & 11 & \cdot & 11 \\ \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & 11 & \cdot & 11 \\ \cdot & \cdot & \cdot & \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ e \\ \hline \cdot \\ e \end{array} \right), \\ &= 11. \end{aligned}$$

En considérant que l'évolution démarre à l'instant 0, il s'agit là bien de l'échéance pour que
- a survienne simultanément au sein de G_1 et G_2 ;
- puis que les séquences ccb et d soient exécutées simultanément (en parallèle) dans G_1 et G_2 ;
- et enfin qu'une nouvelle occurrence de a re-synchronise l'évolution des deux automates.

Il faut remarquer que la séquence ad est achevée au sein de G_2 à l'instant 4 (l'état III est atteint à l'instant 4). Pour autant, la transition d'état consécutive n'a lieu qu'à l'instant 11 (l'évolution de G_2 est en quelque sorte "gelée" durant cet intervalle de temps) car l'événement a n'est alors validé qu'une fois que l'état I est atteint dans G_1.
Notons également que l'ordre entre deux événements privés à des composantes distinctes n'importe pas dans l'évaluation d'une séquence. En effet, comme les événements privés sont supposés s'exécuter de façon asynchrone au sein de chacune des composantes, on obtiendra par exemple la même valeur $l(\mathfrak{G})(w)$ pour les séquences $accbda$, $accdba$, $acdcba$, et $adccba$.

La figure 2.7 est une représentation de l'automate $\mathfrak{G}$ de représentation linéaire (α, μ, β). 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.
Sur la partie droite de la figure 2.6, sont représentés les graphes d'état saufs $\mathcal{G}_1$ et $\mathcal{G}_2$ qui peuvent être associés aux automates G_1 et G_2 (en suivant la démarche proposée dans la deuxième partie de la preuve de la proposition 2.11). L'équivalent de la composition synchrone entre G_1 et G_2

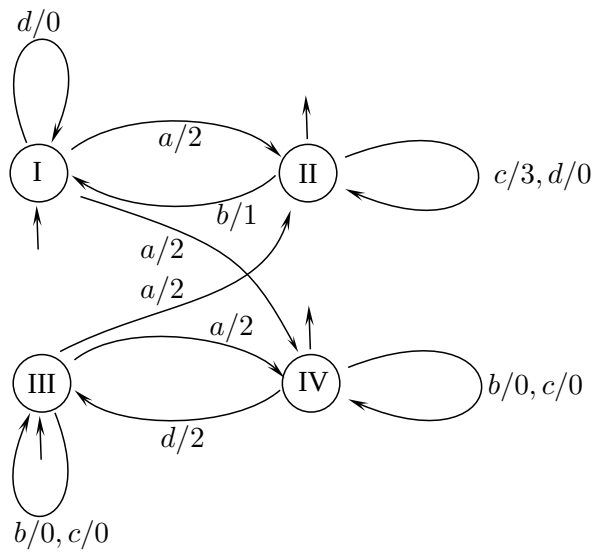

Figure 2.7 : Automate (max,+) $\mathfrak{G}$.

consiste à fusionner les transitions étiquetés par a dans $\mathcal{G}_1$ et $\mathcal{G}_2$, et donc obtenir le réseau de Petri sauf de la figure 2.3. ◇

Remarque 2.15

1. La définition 2.13 s'étend sans difficulté pour pouvoir considérer la composition synchrone de plus de deux automates $(\max, +)$.
2. Pour prouver l'assertion 1, on pourrait montrer que les dates de franchissement des transitions du réseau de Petri sauf obtenu en fusionnant les transitions a de $\mathcal{G}_1$ et $\mathcal{G}_2$ peuvent être évaluées à partir du comportement de $\mathfrak{G}$ [24].
3. A notre connaissance, cette approche pour la composition synchrone d'automates est originale. Il faut souligner que le nombre d'états dans le composé 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., 2009], pour lesquelles le nombre d'états dans l'automate composé est plutôt égal au produit du nombre d'états dans les automates initiaux.
4. On peut se demander s'il existe un automate $(\max, +)$ déterministe équivalent à $\mathfrak{G}$. Il est bien connu (voir par exemple [Cassandras and Lafortune, 2006]) que tout automate [25] 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 Sakarovitch, 2006, Kirsten and Lombardy, 2009]). Pour l'automate $\mathfrak{G}$, ces

24. Pour pouvoir évaluer la durée de l'ensemble des séquences de franchissement possibles dans le réseau de Petri en évaluant le comportement de l'automate $\mathfrak{G}$, il faudrait que tous les états de $\mathfrak{G}$ soient finaux (et donc considérer plutôt l'automate de la figure 2.4). Ici, le calcul $l(\mathfrak{G})(w)$ permet seulement d'évaluer la durée de séquences s'achevant par le franchissement de a dans $\mathcal{G}_1$ et/ou $\mathcal{G}_2$.

25. Pour lequel la fonction de transition prend des valeurs booléennes.

travaux n'apportent pas, à notre connaissance, une réponse immédiate et cela constitue donc une piste à explorer.
D'ici là, on définit dans la section suivante une composition synchrone d'automates $(\max, +)$ sous la forme d'un automate déterministe sur un alphabet étendu.

◇

Produit synchrone résultant en un automate déterministe
On introduit ici une composition synchrone de deux automates $(\max, +)$ sur un alphabet étendu (au lieu de l'alphabet global $A = A_1 \cup A_2$) défini par :

$$\mathsf{A} = (A_1 \cap A_2) \cup (A_1 \setminus A_2)^* \times (A_2 \setminus A_1)^* .$$

Cet alphabet contient les événements communs (dans $A_{com} = A_1 \cap A_2$), et les paires constituées d'une part de l'ensemble des séquences $(A_1 \setminus A_2)^*$ qui peuvent être formées à partir des événements privés de G_1, et d'autre part de l'ensemble des séquences $(A_2 \setminus A_1)^*$ qui peuvent être formées à partir des événements privés de G_2. Cette définition utilise le produit tensoriel de matrice définie sur un dioïde [26] et noté $\otimes^t$. On a également besoin des matrices booléennes associées aux matrices de morphisme :

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

Définition 2.16. *On définit la représentation linéaire* (α, μ, β) *de l'automate* $(\max, +)$ G *sur l'alphabet* A *par*

$$\alpha = \alpha_1 \otimes^t \alpha_2 ,$$

$$\beta = \beta_1 \otimes^t \beta_2 ,$$

et $\mu : \mathsf{A} \to \mathbb{R}_{max}^{|Q| \times |Q|}$ *la matrice de morphisme telle que :*

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

Résumons l'idée sous-jacente à cette définition technique comme suit :
- Les automates G_1 et G_2 se synchronisent lors de l'occurrence d'événements communs $v \in A_1 \cap A_2$. L'expression $\mu_1(v) \otimes^t B_2(v) \oplus B_1(v) \otimes^t \mu_2(v)$ met alors en jeu le produit tensoriel de la matrice de morphisme μ_1 (resp., μ_2) d'une composante avec la matrice booléenne B_2 (resp., B_1) de l'autre composante. Cela permet de transcrire le fait que la transition d'état s'effectue simultanément dans les deux composantes et que la temporisation associée est le maximum des temporisations associées à v.

26. Si A et B sont des matrices de dimensions respectives $m \times n$ et $p \times q$ définies sur un dioïde, alors leur produit tensoriel est la matrice de dimension $mp \times nq$ définie par bloc comme suit

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

– Dans l'intervalle de deux synchronisations, les automates G_1 et G_2 sont libres d'exécuter des séquences d'événements privés représentés par $P_1(v) \in (A_1 \setminus A_2)^*$ et $P_2(v) \in (A_2 \setminus A_1)^*$. L'expression $\mu_1(P_1(v)) \otimes^t B_2(P_2(v)) \oplus B_1(P_1(v)) \otimes^t \mu_2(P_2(v))$ transcrit le fait que ces transitions d'état "privées" s'effectuent de front au sein des composantes, et que la durée associée est le maximum des durées au sein de chacune des composantes.

Remarque 2.17 Si les automates G_i, $i = 1, 2$ sont déterministes, alors
– les vecteurs α_i possèdent un seul coefficient différent de ε (puisqu'il y a un seul état initial),
– pour tout v chaque ligne des matrices $\mu_i(v)$ contient au plus un seul coefficient différent de ε (puisque depuis tout état, au plus une transition par événement est possible).

On peut vérifier que les matrices α et μ de la représentation linéaire de G ont les mêmes propriétés, et donc que l'automate G est déterministe. ◇

Remarque 2.18 Formellement, l'alphabet $\mathsf{A} = (A_1 \cap A_2) \cup (A_1 \setminus A_2)^* \times (A_2 \setminus A_1)^*$ est infini. Dans la pratique, si aucun des automates G_1 ou G_2 ne contient de boucle constituée exclusivement d'événements privés, alors la composition synchrone ne mettra en jeu qu'un nombre fini de paires de séquences privées entre deux synchronisations. ◇

Tout mot défini sur l'alphabet global $w \in (A_1 \cup A_2)^*$ peut être décomposé en des séquences d'événements privés intercalées entre des événements communs, à savoir :

$$w = v_0 a_1 v_1 \dots a_n v_n,$$

où pour chaque $i = 1 \dots n$
- $a_i \in A_1 \cap A_2$,
- $v_i \in (A \setminus (A_1 \cap A_2))^*$.

Pour chaque v_i, $P_1(v_i) \in A_1^*$ et $P_2(v_i) \in A_2^*$ correspondent aux sous-séquences "locales" à chaque composante G_1 et G_2. Tout mot défini sur l'alphabet global $w \in A^*$ peut ainsi s'écrire sur l'alphabet étendu A :

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

Cette décomposition permet de définir la *matrice induite* pour tout mot $w \in A^*$ par :

$$\begin{aligned} \nu(w) \quad = \quad & \mu((P_1(v_0), P_2(v_0)))\ \mu(a_1)\ \mu((P_1(v_1), P_2(v_1))) \dots \\ & \mu(a_n)\ \mu((P_1(v_n), P_2(v_n))) \end{aligned}$$

ainsi que le *comportement induit* par

$$l(\mathsf{G})(w) = \alpha \nu(w) \beta\ .$$

Assertion 2. *La valeur du comportement induit de* G

$$l(\mathsf{G})(w) = \alpha \nu(w) \beta\ ,$$

coïncide avec la date d'achèvement de la séquence w au sein de la composition synchrone de G_1 et G_2.

Il faut souligner que même si l'alphabet étendu A est infini (voir remarque 2.18), la matrice $\nu(w)$ et donc le comportement induit $l(\mathsf{G})(w)$ peuvent être évalués pour tout mot fini $w \in A^*$.

Exemple 2.19 On considère à nouveau les automates représentés sur la figure 2.6. Leurs représentations linéaires sont précisées au début de l'exemple 2.14. Conformément à la définition 2.16, l'automate noté G est défini sur l'alphabet étendu

$$\mathsf{A} = \{a, (b,d), (cb,d), (ccb,d), \ldots\} \subseteq (A_1 \cap A_2) \cup (A_1 \setminus A_2)^* \times (A_2 \setminus A_1)^* \,.$$

L'automate G_1 comporte une boucle constituée d'un événement privé (la transition depuis et vers l'état II sur l'occurrence de c) et l'alphabet étendu est par conséquent infini (remarque 2.18). On a

$$\alpha = \alpha_1 \otimes^t \alpha_2 = \begin{pmatrix} e & \cdot & e & \cdot \end{pmatrix} \,, \beta = \beta_1 \otimes^t \beta_2 = \begin{pmatrix} \cdot \\ e \\ \cdot \\ e \end{pmatrix} \,,$$

$$\begin{aligned}
\mu(a) &= \mu_1(a) \otimes^t B_2(a) \oplus B_1(a) \otimes^t \mu_2(a) \,, \\
&= \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & \varepsilon \end{pmatrix} \otimes^t \begin{pmatrix} \varepsilon & e \\ \varepsilon & \varepsilon \end{pmatrix} \oplus \begin{pmatrix} \varepsilon & e \\ \varepsilon & \varepsilon \end{pmatrix} \otimes^t \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & \varepsilon \end{pmatrix} \,, \\
&= \left(\begin{array}{cc|cc} \cdot & \cdot & \cdot & 2 \\ \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{array}\right) \,,
\end{aligned}$$

et, par exemple,

$$\begin{aligned}
\mu((ccb,d)) &= \mu_1(ccb) \otimes^t B_2(d) \oplus B_1(ccb) \otimes^t \mu_2(d) \,, \\
&= \begin{pmatrix} \varepsilon & \varepsilon \\ 7 & \varepsilon \end{pmatrix} \otimes^t \begin{pmatrix} \varepsilon & \varepsilon \\ e & \varepsilon \end{pmatrix} \oplus \begin{pmatrix} \varepsilon & \varepsilon \\ e & \varepsilon \end{pmatrix} \otimes^t \begin{pmatrix} \varepsilon & \varepsilon \\ 2 & \varepsilon \end{pmatrix} \,, \\
&= \left(\begin{array}{cc|cc} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot \\ 7 & \cdot & \cdot & \cdot \end{array}\right) \,.
\end{aligned}$$

Conformément à l'assertion 2, le calcul de $l(\mathsf{G})(w)$ permet d'établir la date d'achèvement d'un séquence w dans la composition synchrone de G_1 et G_2. Comme dans l'exemple 2.14, nous l'appliquons à la séquence $w = accbda$:

$$\begin{aligned}
l(\mathsf{G})(accbda) &= \alpha\nu(accbda)\beta \,, \\
&= \alpha\mu(a)\mu((ccb,d))\mu(a)\beta \,, \quad . \\
&= 11 \,.
\end{aligned}$$

Ce résultat est identique à celui obtenu dans l'exemple 2.14, et on peut aussi observer l'équivalence avec le réseau de Petri obtenu en fusionnant la transition a des graphes d'état correspondant à G_1 et G_2. $\diamond$

2.2.3 Une approche de commande des automates $(\max, +)$

Les techniques de commande supervisée des systèmes à événements discrets, proposées par Ramadge et Wonham [Ramadge and Wonham, 1989], s'intéressent uniquement à des aspects logiques.
La nécessité de considérer des contraintes temporelles a conduit à étendre ces techniques au cas de systèmes temporisés. On peut notamment citer l'approche de [Brandin and Wonham, 1994] pour laquelle une horloge discrète globale est considérée, ou encore celle de [Alur and Dill, 1994] pour laquelle le temps prend des valeurs continues. Cette dernière approche a été étendue à partir d'abstractions des automates temporisés en automates booléens (logiques), appelés également automates des régions [Wong-Toi and Hoffmann, 1995].
Pour les automates (max,+), on a développé une commande dont différentes facettes sont présentées dans les publications [Komenda et al., 2008, Komenda et al., 2009c, Komenda et al., 2009a, Komenda et al., 2009d, Komenda et al., 2010a]. La suite de ce paragraphe constitue une synthèse de ces travaux.

2.2.3.1 Principe

Le principe de commande proposé pour les automates $(\max, +)$ est proche de celui considéré en *théorie de la commande supervisée* pour les systèmes à évènements discrets modélisés par des automates à états finis [Ramadge and Wonham, 1989].
Le système à événements discrets est représenté par un automate $(\max, +)$ $G = (Q, A, q_0, t, Q_m)$ ayant pour comportement $l(G) \in \overline{\mathbb{R}}_{\max}[\![A]\!]$.
On désire adjoindre un superviseur interagissant avec le système G pour modifier le comportement de ce dernier (c'est-à-dire éventuellement restreindre l'ensemble des séquences reconnues et/ou modifier les durées associées aux séquences reconnues). Pour cela, le superviseur peut conditionner l'occurrence de certains événements. Plus précisément, la partition $A = A_c \cup A_u$ permet de distinguer

- A_c l'ensemble des événements contrôlables : il s'agit d'événements dont le superviseur peut retarder, voire interdire, la validation (et par voie de conséquence, l'occurrence) ;
- A_u l'ensemble des événements incontrôlables : il s'agit d'événements dont le superviseur ne peut ni retarder, ni interdire la validation.

Comme schématisé sur la figure 2.8, le superviseur, noté G_s, interagit comme un retour d'information avec le système G. Le superviseur G_s est un automate (max,+) défini sur le même ensemble d'événements A évoluant en parallèle (de façon synchrone) avec l'automate G du système. A chaque transition d'état au sein de G :

1. le superviseur G_s observe l'événement $a \in A$ survenu dans le système G ;
2. son état évolue en conséquence ;
3. en fonction de l'état atteint, le superviseur spécifie dynamiquement $G_s(a)$ qui traduit l'ensemble des événements contrôlables autorisés jusqu'à l'occurrence d'un nouvel événement dans G, ainsi que le retard de validation imposé par le superviseur pour chacun d'entre eux [27].

L'interaction du superviseur avec le système est formalisée au moyen de la composition parallèle définie comme suit.

Définition 2.20. *[Komenda et al., 2008]*
Soient $G_s = (Q_s, A, q_{s,0}, t_s, Q_{s,m})$ et $G = (Q_g, A, q_{g,0}, t_g, Q_{g,m})$ les automates (max,+) corres-

27. La notation $G_s(a)$ est abusive car G_s n'a pas été spécifié comme une application définie sur l'alphabet A.

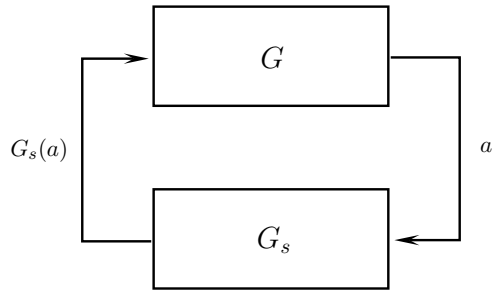

Figure 2.8 : Interaction du superviseur G_s avec le système G sous la forme d'un retour d'information.

pondant respectivement au superviseur et au système. La composition parallèle $G_s \| G$ *de* G *et* G_s*, modélisant le système supervisé, est définie par :*

$$G_s \| G = (Q_s \times Q_g, A, q_0, t, Q_m) \ ,$$

avec

$$\begin{aligned} q_0 &= (q_{s,0}, q_{g,0}), \\ Q_m &= Q_{s,m} \times Q_{g,m}, \end{aligned}$$

$$t((q_s, q_g), a, (q'_s, q'_g)) = \begin{cases} t_s(q_s, a, q'_s) \otimes t_g(q_g, a, q'_g), & si\ a \in A_c, \\ t_g(q_g, a, q'_g), & si\ a \in A_u\ et\ q_s = q'_s, \\ \varepsilon, & si\ a \in A_u\ et\ q_s \neq q'_s. \end{cases} \tag{2.16}$$

On peut se convaincre que la fonction de transition définie ainsi traduit bien l'interaction voulue entre le superviseur et le système. Imaginons que le superviseur (qui observe et évolue de façon synchrone avec le système) atteigne à un instant donné l'état noté q_s :

- Pour un événement contrôlable $a \in A_c$,
  - s'il existe une transition associée à cet événement dans G_s, (*i.e.* $\exists q'_s$ tel que $t_s(q_s, a, q'_s) \neq \varepsilon$), alors cet événement est autorisé par le contrôleur et sa validation dans le système est retardée de $t_s(q_s, a, q'_s)$ unités de temps : dans le système supervisé, la temporisation associée à l'événement est $t_s(q_s, a, q'_s) + t_g(q_g, a, q'_g)$;
  - sinon (*i.e.,* $\nexists q'_s$ tel que $t_s(q_s, a, q'_s) \neq \varepsilon$) l'événement est interdit par le superviseur : la fonction de transition du système supervisé vaut $t_s(q_s, a, q'_s) \otimes t_g(q_g, a, q'_g) = \varepsilon \otimes t_g(q_g, a, q'_g) = \varepsilon$, ce qui traduit qu'aucune transition d'état n'est possible selon l'événement a (même si une transition est possible dans le système G).
- Pour un événement non contrôlable $a \in A_u$, la fonction de transition du système supervisé est donné
  - soit par $t((q_s, q_g), a, (q_s, q'_g)) = t_g(q_g, a, q'_g)$ pour transcrire le fait que la transition a lieu au sein du système sans que le superviseur ne l'influence et que l'action du superviseur ne soit influencée (il n'y a pas de transition dans G_s) ;
  - soit par $t((q_s, q_g), a, (q'_s, q'_g)) = \varepsilon$ si $q_s \neq q'_s$ pour traduire le fait que l'action du superviseur ne doit pas être influencée (qu'il ne doit pas exister de transition associée à cet événement dans G_s).

Remarque 2.21 On peut songer à formaliser l'interaction du superviseur avec le système à l'aide d'un produit synchrone (voir la section 2.2.2.2) pour accentuer l'analogie avec la théorie de la commande supervisée. En considérant que tous les événements sont communs au superviseur et au système, l'idée serait d'affiner la définition en distinguant les événements contrôlables et incontrôlables de sorte à ce que des synchronisations entre le superviseur et le système n'opèrent que sur les événements contrôlables. ◇

2.2.3.2 Formulation du problème de commande

Dans la théorie de la commande supervisée (voir par exemple [Cassandras and Lafortune, 2006, sec. 3.3]), les spécifications, qui précisent l'objectif de commande, sont traduites sous la forme d'un langage admissible. Une procédure (un algorithme) permet de dériver, à partir de ce langage admissible et le langage du système, le langage du superviseur. Un automate reconnaissant ce langage est construit pour jouer le rôle du superviseur.
Là encore, la démarche adoptée pour les automates $(\max, +)$ est comparable. En effet, on se donne une série formelle qui a un rôle équivalent au langage admissible puisqu'elle est censée transcrire le comportement désiré pour le système supervisé. Cette série formelle est conjointement utilisée avec celle du comportement du système pour calculer la série du comportement du superviseur. Le superviseur est finalement obtenu en dérivant un automate $(\max, +)$ reconnaissant cette série formelle en tant que comportement.
Il doit être clair que la synthèse de la commande ne s'appuie pas sur les automates $(\max, +)$ en tant que tels, mais sur les séries formelles associées pour traduire le comportement. Précisons donc le comportement résultant de la composition parallèle du superviseur et du système.

Théoreme 1. *[Komenda et al., 2009d] Le comportement résultant de la composition parallèle de G et G_s est donné par :*

$$l(G_s \| G) = l(G_s) \odot_{A_u} l(G), \tag{2.17}$$

où $\odot_{A_u}$ est appelé produit de Hadamard généralisé *et est défini par*

$$\forall w \in A^*, \quad (l(G_s) \odot_{A_u} l(G))(w) = l(G_s)(P_c(w)) \otimes l(G)(w). \tag{2.18}$$

Le problème de commande traité se résume à résoudre des inégalités définies sur le dioïde [28] de séries formelles $\overline{\mathbb{R}}_{\max}[\![A]\!]$. Plus précisément, étant données des séries formelles de référence (traduisant un comportement désiré pour le système supervisé) y_{ref1} et y_{ref2}, et en notant $y = l(G)$ et $y_s = l(G_s)$ on veut

$$\text{trouver l'ensemble des séries } y_s \text{ telles que } y_{ref1} \succeq y_s \odot_{A_u} y \succeq y_{ref2}\,. \tag{2.19}$$

Ce problème de commande peut aussi se décomposer en :

$$\text{trouver la plus grande série } y_s \text{ telle que } y_{ref1} \succeq y_s \odot_{A_u} y\,, \tag{2.20}$$

et

$$\text{trouver la plus petite série } y_s \text{ telle que } y_s \odot_{A_u} y \succeq y_{ref2}\,. \tag{2.21}$$

28. En raison de l'idempotence de la somme, un dioïde est muni d'une *relation d'ordre partielle*, notée $\preceq$, définie par l'équivalence : $a \preceq b \Leftrightarrow a \oplus b = b$.
En particulier, pour $s, s' \in \overline{\mathbb{R}}_{\max}[\![A]\!]$, $s \preceq s'$ (ordre sur $\overline{\mathbb{R}}_{\max}[\![A]\!]$) correspond à $s(w) \leq s'(w)$ (ordre sur $\overline{\mathbb{R}}_{\max}$ qui coïncide avec l'ordre usuel) pour tout $w \in A^*$. Par exemple, avec $A = \{a, b\}$, $1a \oplus 2ab \preceq 1a \oplus 3ab \oplus 2abb$.

En effet, l'application correspondant au produit de hadamard généralisé est *isotone*[29]. Comme nous montrons de plus que les solutions, respectivement minimale à (2.21) et maximale à (2.20), existent, la solution au problème (2.19) est alors donnée par l'intervalle dont les bornes sont ces solutions extrémales.

Les inégalités (2.20) et (2.21) s'interprètent au regard de l'ordre naturel de $\overline{\mathbb{R}}_{\max}[\![A]\!]$.

- Pour (2.20), il s'agit de trouver la plus grande série y_s, c'est-à-dire, les plus grands coefficients $\overline{y_s}(w)$ pour chaque mot w, et donc (par isotonie du produit de Hamadard généralisé) les plus grands coefficients $(\overline{y_s} \odot_{A_u} y)(w)$, tels que $y_{ref1}(w) \succeq (\overline{y_s} \odot_{A_u} y)(w)$. Ainsi, le superviseur retarde autant que possible l'achèvement des séquences d'événements w au sein du système supervisé (dont le comportement est donné par $\overline{y_s} \odot_{A_u} y$). De plus, comme $y_{ref1} \succeq y_s \odot_{A_u} y$, la date d'achèvement de $(\overline{y_s} \odot_{A_u} y)(w)$ dans le système supervisé est antérieure à celle spécifiée par $y_{ref1}(w)$, ceci pour tout w. Un tel objectif de commande satisfait le critère dit de *juste à temps*, notamment utilisé pour la commande des graphes d'événements temporisés (voir par exemple [Baccelli et al., 1992, §5.6] ou [Lahaye et al., 2008]). D'un point de vue logique, ce superviseur est le plus permissif qui permette de restreindre le comportement du système au comportement maximal souhaitée *via* y_{ref1}.
- Pour (2.21), il s'agit de trouver les plus petits coefficients $\underline{y_s}(w)$ pour chaque w, et donc les plus petits coefficients $(\underline{y_s} \odot_{A_u} y)(w)$, tels que $(\underline{y_s} \odot_{A_u} y)(w) \succeq y_{ref2}(w)$. Autrement dit, le superviseur va ici retarder le moins possible le système. En ce sens, il est à rapprocher de la commande minimisant les retards proposée dans [Houssin et al., 2006a] pour les graphes d'événements temporisés. D'un point de vue logique, ce superviseur est le moins permissif qui garantit le comportement minimal souhaité *via* y_{ref2}.

2.2.3.3 Résultats obtenus

On énonce ici les résultats en occultant les preuves et les aspects techniques : le lecteur intéressé est invité à consulter les références [Komenda et al., 2009c, Komenda et al., 2010a] (la première se trouve en annexe de ce document).

Introduisons la notation $H_y^{A_u}$ pour l'application correspondant au produit de hadamard généralisé, à savoir

$$H_y^{A_u} : s \mapsto s \odot_{A_u} y. \tag{2.22}$$

On a montré que cette application admet des pseudo-inverses, et plus précisément qu'elle est à la fois *résiduable* [Komenda et al., 2009c] et *dualement résiduable*[30] [Komenda et al., 2010a]. Ces résultats conduisent à la proposition suivante pour le problème de commande considéré.

Proposition 2.22. *L'ensemble des solutions au problème (2.19) est donné par l'intervalle de*

29. Une application f est isotone si $a \succeq b$ implique que $f(a) \succeq f(b)$.

30. La *théorie de la résiduation* (voir [Blyth and Janowitz, 1972] pour une présentation exhaustive et [Baccelli et al., 1992, §4.4] pour sa spécification aux dioïdes) permet la définition de 'pseudo-inverses' d'applications isotones . Une application isotone f est dite
- *résiduable* si $\forall y \in \mathcal{C}$, la plus petite borne supérieure de l'ensemble $\{x \in \mathcal{D} | f(x) \preceq y\}$ existe et appartient à cet ensemble (celle-ci est notée $f^\sharp(y)$, l'application $f^\sharp$ étant appelée *résiduée* de f) ;
- *dualement résiduable* si $\forall y \in \mathcal{C}$, la plus grande borne inférieure de l'ensemble $\{x \in \mathcal{D} | f(x) \succeq y\}$ existe et appartient à cet ensemble (celle-ci est notée $f^\flat(y)$, l'application $f^\flat$ étant appelée *résiduée duale* de f).

$\overline{\mathbb{R}}_{\max}[\![A]\!]$ *suivant*

$$\left[(H_y^{A_u})^\flat(y_{ref2}), (H_y^{A_u})^\sharp(y_{ref1})\right].$$

Exemple 2.23 Considérons l'automate (jouant le rôle du système) représenté sur la figure 2.9. On suppose que la validation des événements a, c et d peut être retardée ou même interdite, au contraire de celle de b. On a $A = \{a, b, c, d\}$, avec $A_c = \{a, c, d\}$ et $A_u = \{b\}$.
La représentation linéaire de cet automate G est donnée par

$$\alpha = \begin{pmatrix} e & \varepsilon & \varepsilon \end{pmatrix}, \mu(a) = \begin{pmatrix} \varepsilon & 3 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix}, \mu(b) = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 4 \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix},$$

$$\mu(c) = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 5 & \varepsilon \end{pmatrix}, \mu(d) = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 1 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix}, \beta = \begin{pmatrix} \varepsilon \\ e \\ e \end{pmatrix}.$$

Le comportement de G peut être déduit en calculant

$$y = l(G) = \alpha \otimes \mu^* \otimes \beta = \alpha \otimes (\mu(a)a \oplus \mu(b)b \oplus \mu(c)c)^* \otimes \beta\ ,$$

$$= \alpha \otimes \begin{pmatrix} \varepsilon & 3a & \varepsilon \\ \varepsilon & 1d & 4b \\ \varepsilon & 5c & \varepsilon \end{pmatrix}^* \otimes \beta.$$

On obtient la série de $\overline{\mathbb{R}}_{\max}[\![A]\!]$ suivante :

$$y = 3a(1d \oplus 9bc)^*(4b \oplus e).$$

On souhaite que le système se comporte au plus tard selon la série suivante :

$$y_{ref1} = 7a \oplus 8ad \oplus 9ab \oplus 13abc \oplus 14adb,$$

ce qui signifie que les séquences a, ad, ab, abc et abd devraient être achevées au plus tard aux dates 7, 8, 9, 13 et 14. De plus, toute autre séquence ne devrait pas avoir lieu.
La formule $\overline{y_s} = (H_y^{A_u})^\sharp(y_{ref1})$ permet de calculer la série traduisant le comportement du superviseur le plus permissif en juste-à-temps. On obtient en particulier :

$$\overline{y_c}(a) = 2\ ,\ \ \overline{y_c}(ad) = 4\ ,$$

et

$$\overline{y_c}(w') = \varepsilon,\ \text{pour tout mot } w' \in A_c^*,\ w' \notin \{a, ad\}\ .$$

Le rôle du superviseur dans cet exemple est donc de retarder l'événement (a) de deux unités de temps, d'interdire toute occurrence de (c) et d'autoriser la première occurrence de (d) en la retardant de deux unités de temps puis d'interdire (d). Notons que si la séquence abc (permise dans y_{ref1}) avait été rendue possible, alors la séquence $abcb$ n'aurait pu être empêchée (b étant incontrôlable), ce qui violerait la référence donnée par y_{ref1}.
Le comportement du système avec ce superviseur est $\overline{y_s} \odot_{A_u} y = 5a \oplus 8ad \oplus 9ab \oplus 12adb$.

On souhaite maintenant que le comportement du système supervisé soit supérieur à la série :

$$y_{ref2} = 3a \oplus 4ad \oplus 10adb\ .$$

Cela signifie que l'on souhaite qu'au moins les séquences a, ad et adb se produisent, et qu'elles soient achevées au plus tôt aux dates 3, 4 et 10.

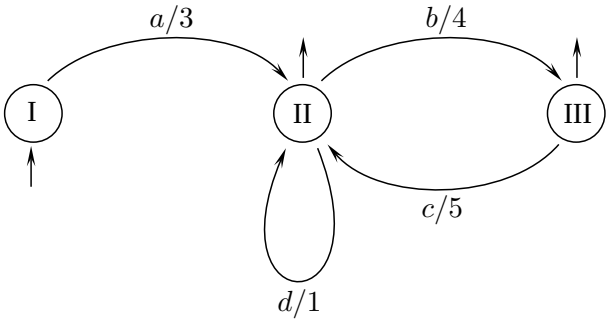

Figure 2.9 : Automate (max,+) modélisant le système.

La formule $\underline{y_s} = (H_y^{A_u})^\flat(y_{ref2})$ permet de calculer la série traduisant le comportement du superviseur le moins permissif en juste-après. On obtient en particulier :

$$\underline{y_s}(a) = 0\ ,\ \ \underline{y_s}(ad) = 2\ ,$$

et

$$\underline{y_s}(w') = \varepsilon,\ \text{pour tout mot } w' \in A_c^*,\ w' \notin \{a, ad\}\ .$$

Le rôle du superviseur dans cet exemple est donc d'autoriser sans retarder l'événement (a), d'interdire toute occurrence de (c) mais d'autoriser la première occurrence de (d) en la retardant de deux unités de temps, puis d'interdire les occurrences suivantes de (d).
Le comportement du système avec ce superviseur est $\underline{y_s} \odot_{A_u} y = 3a \oplus 6ad \oplus 7ab \oplus 10adb$. $\diamond$

Il est possible de spécialiser ces résultats à des séries dites *croissantes sur leur support* (voir [Komenda et al., 2010a]).

Définition 2.24. *Une série* $s \in \overline{\mathbb{R}}_{\max}[\![A]\!]$ *est dite* croissante sur son support *si*

$$\forall v, w \in supp(s), \quad v \preceq w \quad \Rightarrow \quad s(v) \preceq s(w), \tag{2.23}$$

dans laquelle la première inégalité met en jeu la relation d'ordre sur les sous-mots alors que la deuxième inégalité s'entend pour l'ordre naturel de $\overline{\mathbb{R}}_{\max}$.

On peut montrer aisément que le comportement d'un automate $G = (Q, A, \alpha, t, \beta)$ est une série croissante sur son support si, et seulement si, toutes les transitions $t(q, a, q')$ sont positives ou égales à ε. Puisque les valeurs associées aux transitions d'état correspondent à des temporisations, la spécialisation permet donc de se conformer à un cadre réaliste pour les systèmes à événements discrets.

Notons $\overline{\mathbb{R}}_{\max}[\![A]\!]^\uparrow$ le sous-ensemble de $\overline{\mathbb{R}}_{\max}[\![A]\!]$ composé des séries croissantes sur leur support, on cherche, plus précisément, à résoudre le problème de commande de sorte que le superviseur admette pour comportement une série croissante sur son support, c'est-à-dire,

$$\text{trouver } y_c \in \overline{\mathbb{R}}_{\max}[\![A]\!]^\uparrow \text{ tel que } y_{ref1} \succeq y_c \odot_{A_u} y \succeq y_{ref2}. \tag{2.24}$$

On note I l'injection canonique de $\overline{\mathbb{R}}_{\max}[\![A]\!]^\uparrow$ dans $\overline{\mathbb{R}}_{\max}[\![A]\!]$. Cette application est à la fois résiduable et dualement résiduable, et donc l'ensemble des solutions au problème (2.24) est donné par l'intervalle de $\overline{\mathbb{R}}_{\max}[\![A]\!]$ suivant

$$\left[\mathrm{I}^\flat \circ H_y^{A_u\flat}(y_{ref2}), \mathrm{I}^\sharp \circ H_y^{A_u\sharp}(y_{ref1})\right]\ . \tag{2.25}$$

Remarque 2.25 Contrôlabilité La contrôlabilité de la série de référence y_{ref1} a été étudiée dans [Komenda et al., 2009a]. Plus précisément, on a caractérisé les séries qui peuvent être atteintes *via* la commande proposée $(H_y^{A_u})^\sharp(y_{ref1})$. On a notamment proposé une formule permettant de calculer le plus grand "sous-comportement" contrôlable, à savoir $(H_y^{A_u} \circ H_y^{A_u})^\sharp(y_{ref1})$.
$\diamond$

2.2.4 Conclusions et perspectives

On a présenté une démarche visant à faciliter la modélisation de systèmes à événements discrets à l'aide d'automates $(\max, +)$. Elle a pour principe qu'un modèle complexe peut être obtenu de façon modulaire en composant des briques élémentaires modélisant des sous-systèmes simples [31]. L'automate $(\max, +)$ déterministe tient lieu de brique élémentaire, à l'aide de laquelle on modélise de façon intelligible des sous-systèmes mettant en jeu seulement des phénomènes de décalage et de concurrence. Le type de composition entre ces briques, appelée ici *composition synchrone*, revient à considérer qu'il existe des événements communs qui doivent s'exécuter simultanément dans les composantes, et dans l'intervalle desquels des séquences d'événements privés peuvent survenir de façon asynchrone. Deux opérations sont proposées pour traduire la composition synchrone. La première permet d'élaborer un automate $(\max, +)$ défini sur l'union des alphabets des composantes et qui est non-déterministe. Ce produit, original à notre connaissance, est présenté ici pour la première fois. Le deuxième produit synchrone présenté a donné lieu à une communication. Il consiste à construire un alphabet étendu, à partir de ceux des différentes composantes, qui 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. Au moment de l'écriture de ce manuscrit, je ne vois pas de raison pour privilégier l'une ou l'autre des opérations : je suis enclin à penser qu'elles sont complémentaires (que chacune peut trouver un cadre d'utilisation spécifique).

Une approche pour la commande des automates $(\max, +)$ a, en deuxième lieu, été présentée. Le caractère séduisant de ce travail réside dans le fait qu'il concilie le principe de la commande supervisée avec des résultats relevant de la théorie des dioïdes, et qu'il permet de traiter à la fois des aspects logiques et temporels de la commande des systèmes à événements discrets. On peut y opposer l'argument que la classe de systèmes considérés est restreinte. Soulignons notamment qu'un automate $(\max, +)$ déterministe peut être vu comme un automate temporisé avec une seule horloge.

Plusieurs aspects restent à développer. Sans chercher à être exhaustif, on peut en évoquer quelques unes :

- L'approche de commande proposée prend pour objectif des séries de référence à atteindre au mieux. Le choix de ces séries de référence peut requérir un travail d'abstraction conséquent pour l'ingénieur cherchant à appliquer la méthode. Il conviendrait donc de développer une méthodologie qui, à partir de spécifications intelligibles vis-à-vis du système [32], permet la synthèse des séries de référence.
- On a jusque-là considéré que tous les événements peuvent être observés au sein du système. Pour se conformer à un cadre plus réaliste, il est envisagé d'aborder la commande sous observation partielle.

31. Répétons que cette démarche n'est pas originale, et qu'elle est notamment l'écho d'une méthode de modélisation à l'aide de réseaux de Petri.

32. On pense, par exemple, à des spécifications du type : "telle séquence d'événements doit être interdite sur telle période de temps", "les répétitions de telle séquence d'événements doivent être alternée avec telle autre séquence à une cadence voulue", ...

- Pour faire face à la complexité importante et envisager l'application à des systèmes de grande taille, le concept de *commande décentralisée* a été développée en théorie de la commande supervisée. Un travail analogue est à faire pour les automates $(\max, +)$. La définition d'opérations traduisant la composition synchrone peut constituer une première étape dans cette direction, puisqu'elles pourraient être utilisées pour traduire les interactions entre les différentes composantes du dispositif décentralisé.
- En s'appuyant sur la définition proposée dans [Komenda et al., 2010b], on peut envisager une extension des résultats à des systèmes pour lesquels les durées des tâches sont incertaines (ce qui constitue une alternative aux approches stochastiques pour ce type de système).

2.3 Des applications aux systèmes de transport

Les réseaux de transport (réseaux de bus, réseaux ferroviaires,...) et infrastructures routières ont un rôle crucial dans nos sociétés où la circulation des biens et des personnes est prépondérante. Les enjeux associés/inhérents sont multiples et suscitent de nombreux travaux de recherche.
A un certain niveau d'abstraction, il apparaît que la dynamique de ces systèmes est essentiellement régie par des phénomènes, tels que la synchronisation et la concurrence, qui relèvent du formalisme des systèmes à événements discrets[33]. Ce constat a suscité, entre autres, des travaux à partir de modèles reposant sur l'algèbre des dioïdes (sans prétention d'exhaustivité, citons [Braker, 1993, Cohen et al., 2001, Lotito et al., 2005, Goverde, 2007, Farhi et al., 2011]).
Depuis mon recrutement, j'ai également porté un intérêt particulier à ce champ d'applications pour la théorie des systèmes sur l'algèbre des dioïdes :

- Sur la période 2002-2011, j'ai initié et animé une action intitulée *Analyse des réseaux de transports* au LISA (participants : J.-L. Boimond (2002-20011), L. Houssin (2002-2007), I. Demongodin (2003-2006), J. Rousseau (2006-2010)).
- J'ai co-encadré deux Master Recherche/DEA [M1,M3] et deux thèses [T1,T3] sur ce domaine. La suite de cette section précise le contexte et fait une synthèse des travaux développés durant ces thèses.

Lors de ces études, comme dans la suite de cette section, on distingue des applications relatives aux réseaux de transport, puis celles concernant le trafic routier.

2.3.1 Réseaux de transport

On considère que les éléments de base d'un réseau de transport sont les lignes, les arrêts et les entités de transport (bus, tramways ou trains). Ces véhicules parcourent les lignes en embarquant/débarquant des passagers aux arrêts. Le circuit de chaque véhicule est réalisé suivant une planification horaire imposée par une étude préalable prenant notamment en compte le temps de parcours entre les arrêts. Dans ce qui suit, on précise plus encore la manière dont les réseaux de transport sont exploités. Deux phases sont identifiées : une planification (appelée *tableau de marche*) puis le fonctionnement à proprement parler (on parle de *régulation* pour désigner l'ensemble des actions intervenant en opération). Ceci nous permet ensuite de distinguer les contributions à ces deux phases.

33. Par exemple, c'est le cas lorsque la coordination des véhicules au sein d'un réseau est étudiée.

2.3.1.1 Gestion et pratiques d'exploitation

Le *tableau de marche* est une optimisation "hors-ligne" de l'offre de service qui comporte notamment les étapes suivantes (pour une description plus fouillée, le lecteur pourra se référer à [Houssin et al., 2006b, §4.2]) :

- La définition d'une *qualité de service* à fournir : fréquences des passages aux arrêts, répartition des véhicules entre les lignes, correspondances à assurer, ... Celle-ci est définie sur des horizons assez courts (typiquement 2 heures pour les bus) avec une périodicité variable pour s'adapter aux fluctuations des besoins durant la journée (typiquement entre les heures pleines et creuses), durant la semaine (typiquement entre les jours de semaine et le weekend), voire durant l'année (typiquement entre les périodes scolaires et les vacances).

- La synthèse d'une *table d'horaires* qui définit les heures théoriques des passages des véhicules aux arrêts. Cette planification doit, en particulier, répondre aux éléments de qualité de service préalablement décidés.

La *régulation* consiste en des ajustements du tableau de marche réalisées "en-ligne" en réaction aux conditions réelles d'exploitation. Les "manoeuvres de régulation" peuvent être diverses (ignorer ou, au contraire, prolonger certains arrêts, dérouter des véhicules, transférer des passagers, ...) et sont généralement décidées par un superviseur du réseau en interaction avec les chauffeurs. Elles surviennent en particulier, mais pas exclusivement, en cas de perturbation sur le réseau (panne, accident, ...).

2.3.1.2 Modélisation des réseaux de transport opérant selon un tableau de marche

On rappelle une modélisation présentée dans [Houssin et al., 2005, Houssin et al., 2007] et qui se situe dans la continuité de modèles considérés dans [Braker, 1993, Goverde, 2007]. Il est pris pour hypothèse que les véhicules desservant une même ligne ne se dépassent pas, et les variables associées au système de transport sont des dateurs :

- $x_i(k)$, $k \in \mathbb{Z}$, désigne la date du $k+1$-ième départ d'un véhicule depuis l'arrêt indexé i du réseau.

Des phénomènes de décalage et de synchronisation opèrent au sein d'un réseau :

- Si on désigne par a_{ij} le temps de parcours entre deux arrêts successifs j et i, alors on a[34]

$$x_i(k) \geq a_{ij} + x_j(k) \,.$$

- Si l'arrêt i est en correspondance avec l'arrêt l et que l'on note a_{il} le temps de transit des passagers entre ces arrêts[35], alors on a

$$x_i(k) \geq a_{il} + x_l(k) \,.$$

- La table d'horaires est modélisée par un vecteur $u(k)$: $u_i(k)$ désigne la date de départ prévue pour le $k+1$-ième véhicule depuis l'arrêt indexé i. On a

$$x_i(k) \geq u_i(k) \,.$$

34. On considère qu'à l'initialisation aucun véhicule ne démarre entre les deux arrêts. Cette hypothèse vise simplement à faciliter l'exposé et peut être levée sans difficulté.

35. On considère qu'un véhicule ne peut partir de l'arrêt i que s'il a pu embarquer des passagers débarqués à l'arrêt l.

En supposant que le système fonctionne au plus tôt [36], et par des manipulations combinatoires (identiques à celles présentées dans [Baccelli et al., 1992, §2.5]), on peut se ramener à une équation d'état $(\max, +)$-linéaire standard, à savoir :

$$x(k) = A \otimes x(k-1) \oplus B \otimes u(k) \ , \tag{2.26}$$

où x et u sont les vecteurs des dateurs associés aux arrêts et à la table d'horaires.

En outre, une partie de la *qualité de service* attendue dans le réseau peut être exprimée sous la forme de contraintes sur les variables d'état du système.

- Des échéances peuvent être fixées pour les départs à certains arrêts. Notant ξ_i le dateur des départs au plus tard [37] depuis l'arrêt i, cette exigence se traduit par l'inégalité

$$\forall k,\ x_i(k) \leq \xi_i(k) \ . \tag{2.27}$$

- Il est usuel d'imposer un intervalle de temps minimum [38] entre deux départs à un même arrêt [39], c'est-à-dire de contraindre

$$\forall k,\ x_i(k+1) \geq \Delta_i^{min} + x_i(k) \ , \tag{2.28}$$

  où Δ_i^{min} est l'intervalle minimum souhaité à l'arrêt i.
- Pour assurer une fréquence minimum de passage, on peut imposer un intervalle de temps maximum entre deux départs, à savoir

$$\forall k,\ x_i(k+1) \leq \Delta_i^{max} + x_i(k) \ . \tag{2.29}$$

- A un arrêt du réseau, un nombre voulu de départs peut être exigé entre des dates données [40]. Cette contrainte s'exprime par

$$\exists k \in \mathbb{Z} \text{ tel que } x_i(k) \geq t_i \text{ et } x_i(k+m) \leq t_i + r \ , \tag{2.30}$$

  où m est le nombre minimum de départs devant avoir lieu depuis l'arrêt i sur l'intervalle de dates $[t_i, t_i + r]$.
- Il peut être souhaité de borner le temps de parcours sur une partie d'une ligne. En notant δ_{ji} le temps de parcours maximum souhaité depuis l'arrêt i jusqu'à l'arrêt j, cette contrainte s'exprime [41]

$$\forall k,\ x_j(k) \leq \delta_{ji} + x_i(k) \ . \tag{2.31}$$

- Il peut être souhaité de borner le nombre de véhicules sur une partie d'une ligne. En notant κ_{ji} le nombre maximum de véhicules tolérés sur le tronçon commençant par l'arrêt i et se terminant par l'arrêt j, cette contrainte s'exprime[41].

$$\forall k,\ x_j(k - \kappa_{ji}) \leq x_i(k) \ . \tag{2.32}$$

36. Ceci signifie que les véhicules démarrent aux arrêts dès que possible.
37. Autrement dit, le départ du $k+1$-ième véhicule depuis un arrêt i doit avoir lieu avant la date $\xi_i(k)$.
38. Dans la littérature en langue anglaise, le terme utilisé est *minimum headway*.
39. Cette contrainte est particulièrement pertinente pour les réseaux de bus afin d'éviter la formation de pelotons de véhicules (plusieurs bus se suivant sur une même ligne). Un tel phénomène a naturellement tendance à se produire, notamment dès qu'un véhicule est légèrement retardé : il a plus de passagers à embarquer et les véhicules qui le suivent ont tendance à le rattraper.
40. Par exemple, en prévision de pics d'affluence de passagers à des heures et à des arrêts connus du réseau (fin des classes à proximité d'une école, fin de journée aux abords d'une usine,...).
41. On considère qu'à l'initialisation aucun véhicule ne démarre entre les deux arrêts. Cette hypothèse vise simplement à faciliter l'exposé et peut être levée sans difficulté.

2.3.1.3 Contribution au tableau de marche

La synthèse d'une table d'horaires est la phase finalisant le tableau de marche d'un réseau de transport. Une approche basée sur l'algèbre $(\max, +)$ a été proposée dans [Houssin et al., 2005, Houssin et al., 2007] pour cette étape.

Le problème considéré peut se résumer comme suit. Sur un horizon temporel choisi [42], on se donne

- des temps moyens de parcours et de transit (autrement dit les valeurs des coefficients de la matrice A dans (2.26)),
- une qualité de service désirée (décrite par des inéquations (2.27)-(2.31)),

et on veut synthétiser la table d'horaires du réseau (c'est-à-dire, calculer un dateur $\{u(k)\}_{k\in\mathbb{Z}}$ tel que l'état x régi par (2.26) satisfasse les inéquations (2.27)-(2.31)).

Du point de vue de l'automaticien, le problème considéré est un problème de commande en boucle ouverte, et plus précisément un problème de poursuite de la trajectoire $\{\xi(k)\}_{k\in\mathbb{Z}}$ dans (2.27) tout en satisfaisant des contraintes s'écrivant sous les formes (2.28)-(2.31).

La démarche proposée dans [Houssin et al., 2005, Houssin et al., 2007] consiste à formuler l'ensemble des contraintes (2.27)-(2.31) traduisant la qualité de service désirée sous la forme d'une inéquation dans $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ [43] sur le vecteur de commande [Houssin et al., 2007, § 4.3], à savoir

$$u \preceq f(u)\,, \tag{2.33}$$

où f est l'application traduisant l'ensemble des contraintes considérées.
Une méthode itérative peut alors être appliquée pour calculer la plus grande trajectoire u satisfaisant (2.33) [Houssin et al., 2007, prop. 4] [44]. La démarche a été appliquée pour la synthèse de la table d'horaires pour deux lignes de bus du réseau de la ville d'Angers [Houssin et al., 2007, § 4.5] (une version étendue de l'exemple peut être trouvée dans [Houssin, 2006, § 4.5]).

Établir la plus grande trajectoire u vérifiant (2.33) permet de disposer des dates les plus tardives de départs des véhicules tout en satisfaisant les objectifs de qualité de service. L'optimalité de la solution est pertinente du point de vue de l'exploitant puisqu'elle permet de limiter le nombre de véhicules roulant dans le réseau.

2.3.1.4 Contribution à la régulation

La régulation correspond à des adaptations du tableau de marche décidées "en-ligne" pour faire face à des écarts entre les conditions moyennes considérées pour la synthèse de la table d'horaires et les conditions courantes d'exploitation. On résume ici les travaux proposés dans [Houssin et al., 2006a] et élaborés dans l'optique de répondre à des problématiques de régulation de réseau de transport. Ces résultats de commande des systèmes $(\max, +)$-linéaires ont en effet été motivés par une application à la régulation d'un noeud ferroviaire.

Le problème considéré peut se résumer comme suit. On suppose un réseau fonctionnant selon son tableau de marche, à savoir un réseau dont le fonctionnement peut être modélisé par une équation d'état $(\max, +)$-linéaire (2.26) dans laquelle le signal d'entrée u est connu (table d'horaires synthétisée au préalable). En fonction des dates de départ effectives (mesures

42. Correspondant à une période durant laquelle le trafic et les besoins sont stables.

43. L'analogue de la transformée en $\mathcal{Z}$ (utilisée pour les signaux en temps discret) peut être introduite pour les dateurs. Cette transformée permet de manipuler des séries formelles x (en deux variables commutatives γ et δ et à exposants dans $\mathbb{Z}$) représentant les trajectoires des dateurs $\{x(k)\}_{k\in\mathbb{Z}}$. L'ensemble de ces séries formelles est un dioïde noté $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ (la construction de ce dioïde est détaillée dans [Baccelli et al., 1992, chap. 5])

44. La loi de commande ainsi établie a été qualifiée de *commande en juste-à-temps sous contraintes* et il s'agit d'une évolution d'une commande proposée dans [Cohen et al., 1989] et étendue dans [Menguy et al., 2000].

des variables d'état du système x), la régulation envisagée consiste à éventuellement retarder certains départs de véhicules. Plus précisément, dans l'équation d'état modélisant le réseau

$$x(k) = A \otimes x(k-1) \oplus B \otimes u(k) \,, \tag{2.34}$$

on propose de remplacer $\{u(k)\}_{k\in\mathbb{Z}}$ qui code les horaires de départ prévus dans la table d'horaires par une nouvelle entrée $\{u'(k)\}_{k\in\mathbb{Z}}$ donnée par l'expression suivante dans $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$:

$$u' = u \oplus F \otimes x \,, \tag{2.35}$$

dans laquelle F est une matrice de pondération à déterminer et la trajectoire x correspond aux dates de départs mesurées dans le réseau en fonctionnement. Le dateur u' traduit alors les dates de départ des véhicules décidées en exploitation (typiquement au sein d'un poste de supervision et transmises en temps-réel aux chauffeurs). Notons que puisque $\forall k$, $u'(k) \geq u(k)$, la régulation consiste à éventuellement retarder les départs au regard des horaires prévus *via* u [45]. En l'occurrence, une prolongation de l'arrêt d'un véhicule est décidée à un arrêt i si $u_i \preceq (F \otimes x)_i$, c'est-à-dire, si un retard sur l'horaire planifié est apparu dans le réseau. Il peut paraître discutable, voire contestable, d'envisager de retarder plus encore des véhicules qui ont pris du retard. Nous justifions cette approche en considérant que des préoccupations prioritaires (relevant, par exemple, de la sécurité) vis-à-vis de la qualité de service peuvent exister. Le problème de commande est ainsi complété en considérant des contraintes additionnelles (traduisant ces exigences prioritaires ne relevant pas de la qualité de service) sous la forme d'une inégalité dans $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$:

$$x \succeq g(x) \,. \tag{2.36}$$

Des contraintes sous les formes (2.28), (2.31), (2.32) peuvent notamment être considérées pour, par exemple, garantir un nombre maximum de véhicules et/ou un temps de parcours maximum sur certains tronçons du réseau (une fois de plus, en imaginant que ces exigences soient cruciales au point d'envisager de dégrader la qualité de service pour les satisfaire). L'objectif de commande considéré est alors de minimiser autant que possible les retards tout en satisfaisant ces contraintes. Plus formellement (voir [Houssin et al., , Houssin et al., 2006a, Houssin, 2006]), le problème posé est de calculer la plus petite matrice F telle que, pour toute table d'horaires u, l'état du système obtenu en appliquant l'entrée (2.35) satisfasse les contraintes additionnelles (2.36). Du point de vue de l'automaticien, il s'agit d'un problème de synthèse d'un plus petit correcteur F de type retour d'état [46]. Il s'avère que ce problème peut se ramener à la recherche d'un plus petit point pré-fixe d'une application antitone, et on a proposé une méthode itérative permettant d'obtenir une solution au problème de commande approchant une solution minimale [Houssin et al., 2006a].

2.3.2 Trafic routier

Les modèles pour le trafic routier sont généralement classés selon trois échelles de représentation : microscopique, mésoscopique et macroscopique. Un modèle microscopique a un niveau de détail très élevé et l'entité est le véhicule. Un tel modèle décrit aussi bien le comportement espace-temps des véhicules que leurs interactions individuelles. C'est aussi le cas pour les modèles mésoscopiques à la différence que l'entité correspond à des groupes de véhicules. Les modèles

45. Soulignons qu'un seul type de manoeuvre de régulation est donc envisagé.

46. Cette loi de commande a été qualifiée de *commande sous contraintes en minimisant les retards*. Elle est, à notre connaissance, elle est originale à la fois pour le critère considéré et la méthode de synthèse proposée.

macroscopiques décrivent le comportement du trafic routier par des variables agrégées telles que le débit, la concentration et la vitesse moyenne du flux de véhicules.

Différents travaux basés sur l'algèbre $(\min, +)$ ont été proposés pour modéliser et étudier le trafic routier. Dans [Lotito et al., 2005, Farhi et al., 2007], les auteurs proposent un modèle microscopique basé sur les réseaux de Petri qui consiste à diviser la route en une suite de tronçons (correspondant chacun à une place de réseau de Petri) ne pouvant contenir qu'un véhicule à la fois[47]. Puis une modélisation matricielle de ces réseaux de Petri dans l'algèbre $(\min, +)$, conduit à interpréter le calcul des flux moyens comme des problèmes de valeurs propres généralisées. L'idée est donc d'étudier les mouvements microscopiques des véhicules pour dériver des lois macroscopiques. Les auteurs ont en particulier porté leur attention sur le cas d'une route circulaire, de deux routes avec une intersection et sur des réseaux réguliers de routes.

Dans les travaux de thèse de J. Rousseau [Rousseau, 2010], on a étudié le trafic routier sur un tronçon en le considérant comme un système $(\min, +)$-linéaire et en le modélisant à l'aide de la réponse impulsionnelle associée. L'intérêt est de fournir un modèle dont on peut à notre guise fixer l'échelle de représentation et ainsi

- capturer aussi bien des phénomènes microscopiques que macroscopiques ;
- s'adapter au cantonnement existant du fait des dispositifs de mesure en place sur certaines routes. On fait ici référence aux outils en vigueur pour l'information et l'étude du trafic qui reposent sur des détecteurs (souvent des boucles électro-magnétiques) eux-mêmes connectés à des Unités de Mesure et de Traitement (UMT) qui permettent d'enregistrer différentes informations sur chaque véhicule circulant (horodatage du passage, vitesse, longueur, . . .) et d'établir des indicateurs (vitesse, moyenne, . . .). La répartition géographique de ces dispositifs permet de voir une infrastructure comme un ensemble de tronçons (délimités par ces détecteurs).

La contribution de la thèse concerne l'identification des systèmes linéaires dans l'algèbre $(\min, +)$, et consiste en des méthodes d'estimation des paramètres d'applications permettant d'approcher la réponse impulsionnelle de systèmes $(\min, +)$-linéaires. L'objectif du travail a été de fournir des outils qui, à partir de mesures de trafic sur un tronçon routier (qui peuvent être celles transmises par les détecteurs/UMT en vigueur), permettent de déduire un modèle min-plus linéaire de celui-ci. Une méthodologie est également ébauchée pour composer entre eux les modèles de tronçons et ainsi disposer de modèles pour des infrastructures routières complexes.

2.3.3 Conclusion et perspectives

Les travaux présentés dans cette section ont contribué, de mon point de vue, à démontrer la propension de la théorie des systèmes sur les dioïdes à répondre à des questions d'intérêt pour l'étude et l'exploitation des systèmes de transport. Ce qui, de plus, est particulièrement satisfaisant est que ces travaux à des fins d'application ont fait émerger des problématiques originales au sein de cette théorie. La synthèse de correcteur minimisant les retards, présentée dans le paragraphe 2.3.1 et motivée par la régulation d'un noeud ferroviaire, en est un exemple. On peut nous reprocher de ne pas avoir mené la démarche jusqu'au bout, c'est-à-dire jusqu'à l'application, ou encore, jusqu'à l'activité de transfert pour mettre en oeuvre ces travaux sur des systèmes de transport réels. Par exemple, la méthode proposée pour le calcul des tables d'horaires mériterait d'être confrontée aux outils (reposant sur de la programmation linéaire en variables mixtes) utilisés par les exploitants de réseau.
Concernant les perspectives, les travaux initiés sur le trafic routier ne sont clairement pas aboutis

47. Comme dans les *automates cellulaires*, autre modèle microscopique largement utilisé.

et mériteraient d'être poursuivis. A mon sens, il est pertinent d'appliquer des techniques d'identification des systèmes linéaires sur les dioïdes pour modéliser le trafic routier sur un tronçon de route. Néanmoins, les hypothèses qui ont jusque-là été considérées sont restrictives, et on ne peut conséquemment pas envisager d'application à des infrastructures complexes. Il conviendrait donc de généraliser ces travaux.

Chapitre 3

Conclusions et perspectives

3.1 Conclusions

Le chapitre 1 présente mon implication dans l'enseignement supérieur depuis mon recrutement en 2001 à l'Université d'Angers. J'y ai assuré la charge d'enseignement relevant d'un maître de conférences, avec une prise de responsabilité progressive dans les instances et la vie de cet établissement.

Le chapitre 2 est une synthèse des travaux de recherche auxquels je contribue depuis une dizaine d'années. Ils ont été menés principalement dans le cadre de quatre thèses co-encadrées (dont une en-cours) et une collaboration active avec un collègue étranger. On a souligné les lignes directrices communes à ces travaux pour faire apparaître leur cohérence scientifique. Je dois reconnaître que, pour autant, ces travaux ne constituent pas un "monolithe", car il s'agit plutôt d'un ensemble de contributions relatives à différentes problématiques sur les systèmes à événements discrets.

3.2 Perspectives

Au moment de l'écriture de ce bilan, les travaux sur les automates $(\max, +)$ constituent l'axe de recherche que je compte prioritairement poursuivre. Il est à cheval sur deux théories [1] qui concourent à l'étude des systèmes à événements discrets. Ces travaux "à l'interface" permettent de mieux cerner les atouts et complémentarités des deux approches. Le lecteur aura constaté que les résultats sur ce sujet ont été, en grande partie, obtenus dans le cadre d'une collaboration particulièrement fructueuse et enrichissante qui va se poursuivre avec Jan Komenda. La connexité de nos travaux avec ceux d'autres collègues français et étrangers laissent à penser que le cercle des partenaires sur ce sujet pourrait être agrandi. Cela constituerait notamment l'opportunité, sans aucun doute enrichissante, de côtoyer plus encore des collègues appartenant à la communauté informatique.

De par ma formation en ingénierie et ma fonction d'enseignant dans une école d'ingénieur, mon travail de recherche est forcément empreint, inspiré, voire motivé par les applications potentielles. Participant à la formation d'ingénieurs qui conçoivent et optimisent des systèmes

1. La *théorie des systèmes sur les dioïdes* et la *théorie de la commande supervisée*.

automatisés, j'ai notamment l'espoir que les études sur les systèmes à événements discrets aboutissent à des méthodes et outils employés par de tels ingénieurs. L'intérêt particulier que je porte aux applications pour les systèmes de transports va se poursuivre. J'ai, en particulier, l'intention d'approfondir nos premiers travaux sur le trafic routier. Il s'agit d'un sujet de recherche actif au sein de la communauté : on pense notamment aux travaux de Nadir Farhi et Jean-Pierre Quadrat ainsi qu'aux nombreux travaux basés sur les réseaux de Petri. Des interactions fructueuses sont donc certainement possibles, et je compte par la suite œuvrer dans ce sens.

Bibliographie

[Alsaba et al., 2006a] Alsaba, M., Boimond, J.-L., and Lahaye, S. (2006a). Sur la commande des systèmes flexibles de production manufacturière par l'algèbre des dioïdes. In *Conférence Internationale Francophone d'Automatique (CIFA 2006)*, Bordeaux, France.

[Alsaba et al., 2007] Alsaba, M., Boimond, J.-L., and Lahaye, S. (2007). Sur la commande des systèmes flexibles de production manufacturière par l'algèbre des dioïdes. *Sciences et Technologies de l'Automatique (e-STA)*, vol. 4(2).

[Alsaba et al., 2006b] Alsaba, M., J-L.Boimond, and S.Lahaye (2006b). On just in time control of flexible manufacturing systems via dioid algebra. In *Information Control Problems in Manufacturing (INCOM 2006)*, volume 2, pages 137–142, Saint-Etienne, France.

[Alsaba et al., 2006c] Alsaba, M., Lahaye, S., and Boimond, J.-L. (2006c). On just in time control of switching max-plus linear systems. In *International Conference on Informatics in Control, Automation and Robotics(ICINCO 2006)*, setubal, Portugal.

[Alur and Dill, 1994] Alur, R. and Dill, D. (1994). A theory of timed automata. *Theoretical Computer Science*, 126 :183–235.

[Baccelli et al., 1992] Baccelli, F., Cohen, G., Olsder, G.-J., and Quadrat, J.-P. (1992). *Synchronization and Linearity.* Wiley.

[Berthomieu and Diaz, 1991] Berthomieu, B. and Diaz, M. (1991). Modeling and Verification of Time Dependent Systems using Time Petri Nets. *IEEE Transaction on Software Eng.*, vol. 17(3) :259–273.

[Blyth and Janowitz, 1972] Blyth, T. S. and Janowitz, M. F. (1972). *Residuation Theory.* Pergamon press.

[Boimond and Ferrier, 1996] Boimond, J.-L. and Ferrier, J.-L. (1996). Internal Model Control and Max-Algebra : Controller Design. *IEEE Trans. on Automatic Control*, 41(3) :457–461.

[Braker, 1993] Braker, H. (1993). *Algorithms and Applications in Timed Discrete Event Systems.* PhD thesis, Delft University of Technology.

[Brandin and Wonham, 1994] Brandin, B. A. and Wonham, W. M. (1994). Supervisory control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, 39 :329–342.

[Buchholz and Kemper, 2003] Buchholz, P. and Kemper, P. (2003). Weak bisimulation for (max,+) automata and related models. *J. Autom. Lang. Comb.*, 8 :187–218.

[Cassandras and Lafortune, 2006] Cassandras, C. G. and Lafortune, S. (2006). *Introduction to Discrete Event Systems.* Springer-Verlag New York, Inc.

[Cohen et al., 2001] Cohen, G., Gaubert, S., Mancinelli, E., Quadrat, J.-P., and Rofman, E. (2001). On traffic light control of regular towns. *Rapport INRIA n. 4276.*

[Cohen et al., 1984] Cohen, G., Moller, P., Quadrat, J. P., and Viot, M. (1984). Linear system theory for discrete-event systems. In *23rd IEEE Conf. on Decision and Control*, Las Vegas, Nevada.

[Cohen et al., 1986] Cohen, G., Moller, P., Quadrat, J.-P., and Viot, M. (1986). Dating and counting events in discrete event systems. In *25th IEEE Conf. on Decision and Control*, Athens, Greece.

[Cohen et al., 1989] Cohen, G., Moller, P., Quadrat, J. P., and Viot, M. (1989). Algebraic tools for the performance evaluation of discrete event systems. *IEEE Proceedings : Special issue on Discrete Event Systems*, 77(1).

[Cottenceau et al., 2001] Cottenceau, B., Hardouin, L., Boimond, J.-L., and Ferrier, J.-L. (2001). Model Reference Control for Timed Event Graphs in Dioids. *Automatica*, vol. 37 :1451–1458.

[David and Alla, 1989] David, R. and Alla, H. (1989). *Du Grafcet aux réseaux de Petri*. Hermès, Paris.

[De Schutter and van den Boom, 2001] De Schutter, B. and van den Boom, T. (2001). Model predictive control for max-plus-linear discrete event systems. *Automatica*, 37(7) :1049 – 1056.

[De Schutter and van den Boom, 2003] De Schutter, B. and van den Boom, T. (2003). MPC for discrete-event systems with soft and hard synchronisation constraints. *International Journal of Control*, 76(1) :82–94.

[Farhi et al., 2007] Farhi, N., Goursat, M., and Quadrat, J.-P. (2007). Road traffic models using petri nets and min-plus algebra. In *Conf. on Traffic and Granular Flow*, Orsay, France.

[Farhi et al., 2011] Farhi, N., Goursat, M., and Quadrat, J.-P. (2011). The traffic phases of road networks. *Transportation Research C*, 19 :85–102.

[Gaubert, 1992] Gaubert, S. (1992). *Théorie des systèmes linéaires dans les dioïdes*. Thèse de doctorat, Ecole des Mines de Paris.

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

[Gaubert and Mairesse, 1999] 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.

[Gaujal, 1994] Gaujal, B. (1994). *Parallélisme et simulation des systèmes à événements discrets*. Thèse de doctorat, Université de Nice – Sophia Antipolis.

[Goverde, 2007] Goverde, R. (2007). Railway timetable stability analysis using max-plus system theory. *Transportation Research Part B*, 2 :179–201.

[Hamaci et al., 2006a] Hamaci, S., Boimond, J.-L., and Lahaye, S. (2006a). Modeling and Control of Hybrid Timed Event Graphs with Multipliers using (Min,+) Algebra. *Discrete Event Dynamic Systems*, vol. 16(2) :241–256.

[Hamaci et al., 2006b] Hamaci, S., Boimond, J.-L., and Lahaye, S. (2006b). *Part 3-Signal Processing, Systems Modeling and Control. On Modeling and Control of Discrete Timed Event Graphs with Multipliers using (Min,+) Algebra*. SPRINGER. J. Bras et al. (Editors), Informatics in Control, Automation and Robotics I. ISBN 1-4020-4136-5.

[Hamaci et al., 2007] Hamaci, S., Boimond, J.-L., and Lahaye, S. (2007). *Part 3-Signal Processing, Systems Modeling and Control. Performance Analysis of Timed Event Graphs with Multipliers using (Min, +) Algebra*. SPRINGER. J. Filipe et al. (Editors), Informatics in Control, Automation and Robotics II. ISBN 1-4020-5625-3.

[Heidergott et al., 2006] Heidergott, B., Olsder, G. J., and Woude, J. V. D. (2006). *Max Plus at work*. Princeton.

[Houssin, 2006] Houssin, L. (2006). *Contribution à la commande des systèmes (max,+)-linéaires. Aplications aux réseaux de transport.* Thèse de doctorat, Université d'Angers.

[Houssin et al.,] Houssin, L., Lahaye, S., and Boimond, J.-L. Control of (max,+)-linear systems minimizing delays. *Discrete Event Dynamic Systems.* accepté sous la réserve de modifications mineure en cours.

[Houssin et al., 2005] Houssin, L., Lahaye, S., and Boimond, J.-L. (2005). Commande en juste-à-temps sous contraintes de systèmes (max,+)-linéaires. In *Modélisation des Systèmes Réactifs (MSR 2005)*, volume 39, pages 335–350, Grenoble, France. publié dans Special issue of Journal Européen des Systèmes Automatisés (JESA).

[Houssin et al., 2006a] Houssin, L., Lahaye, S., and Boimond, J.-L. (2006a). Control Of Constrained (max,+)-Linear Systems Minimizing Delays. In *Information Control Problems in Manufacturing (INCOM 2006)*, Saint-Etienne, France.

[Houssin et al., 2006b] Houssin, L., Lahaye, S., and Boimond, J.-L. (2006b). Timetable Synthesis Using (max,+) Algebra. In *Information Control Problems in Manufacturing (INCOM 2006)*, Saint-Etienne, France.

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

[Kirsten and Lombardy, 2009] 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)*, pages 589–600.

[Klimann et al., 2004] 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 et al., 2008] Komenda, J., Boimond, J.-L., and Lahaye, S. (2008). Control of (max,+) automata : logical and timing aspects. In *9th international workshop on discrete event systems (WODES 2008)*, pages 55–60, Göteborg, sweden.

[Komenda et al., 2009a] Komenda, J., Lahaye, S., and Boimond, J.-L. (2009a). Controllability of (max,+) Formal Power Series. In *IFAC Workshop on Dependable Control of Discrete Systems (DCDS 2009)*, Bari, Italy.

[Komenda et al., 2009b] Komenda, J., Lahaye, S., and Boimond, J.-L. (2009b). Le produit synchrone des automates (max,+). *Special issue of Journal Européen des Systèmes Automatisés (JESA)*, vol. 43(7) :1033–1047.

[Komenda et al., 2009c] Komenda, J., Lahaye, S., and Boimond, J.-L. (2009c). Supervisory Control of (max,+) Automata : A Behavioral Approach. *Discrete Event Dynamic Systems*, vol. 19(4) :525–549.

[Komenda et al., 2009d] Komenda, J., Lahaye, S., and Boimond, J.-L. (2009d). Supervisory Control of (max,+) Automata : A Single Step Approach. In *European Control Conference (ECC'09)*, Budapest, Hongrie.

[Komenda et al., 2010a] Komenda, J., Lahaye, S., and Boimond, J.-L. (2010a). Commande supervisée des automates (max,+). In *Conférence Internationale Francophone d'Automatique (CIFA 2010)*, Nancy, France.

[Komenda et al., 2010b] Komenda, J., Lahaye, S., and Boimond, J.-L. (2010b). Synchronous Composition of Interval Weighted Automata. In *International Workshop on Discrete Event Systems (WODES 2010)*, Berlin, Allemagne.

[Lahaye et al., 2008] Lahaye, S., Boimond, J.-L., and Ferrier, J.-L. (2008). Just in Time Control of Time-Varying Discrete Event Dynamic Systems in (max,+) Algebra. *International Journal of Production Research (IJPR)*, vol. 46(19) :5337–5348.

[Lahaye et al., 2004] Lahaye, S., Boimond, J. L., and Hardouin, L. (2004). Linear Periodic Systems Over Dioids. *Discrete Event Dynamic Systems*, vol. 14 :133–152.

[Lhommeau et al., 2004] Lhommeau, M., Hardouin, L., Cottenceau, B., and Jaulin, L. (2004). Interval analysis in diod : Application to robust controller design for Timed Event Graphs. *AUTOMATICA*, vol. 40 :1923–1930.

[Libeaut, 1996] Libeaut, L. (1996). *Sur l'utilisation des dioïdes pour la commande des systèmes à événements discrets*. Thèse de doctorat, Ecole Centrale de Nantes.

[Lombardy and Sakarovitch, 2006] Lombardy, S. and Sakarovitch, J. (2006). Sequential ? *Theoretical Computer Science*, 359(1-2) :224–244.

[Lotito et al., 2005] Lotito, P., Mancinelli, E., and Quadrat, J.-P. (2005). A minplus derivation of the fundamental car-traffic law. *IEEE Transactions on Automatic Control*, 50 :699–705.

[Maia et al., 2003] Maia, C., Hardouin, L., Mendes, R. S., and Cottenceau, B. (2003). Optimal Closed-Loop Control for Timed Event Graphs in Dioid. *IEEE Transaction on Automatic Control*, vol. 48 :2284–2287.

[Menguy et al., 2000] Menguy, E., Boimond, J.-L., Hardouin, L., and Ferrier, J.-L. (2000). Just in time control of linear systems in dioid : cases of an update of reference input and uncontrollable input. *IEEE Transaction on Automatic Control*, vol. 45(11) :2155–2159.

[Moller, 1988] Moller, P. (1988). *Théorie algébrique des Systèmes à Événements Discrets*. Thèse de doctorat, École des Mines de Paris.

[Murata, 1989] Murata, T. (1989). Petri Nets : Properties, Analysis and Applications. *IEEE Proceedings : Special issue on Discrete Event Systems*, 77 :541–580.

[Ramadge and Wonham, 1989] Ramadge, P. J. G. and Wonham, W. M. (1989). The control of discrete event systems. *Proceedings of The IEEE*, 77 :81–98.

[Ramchandani, 1973] Ramchandani, C. (1973). *Analysis of asynchronous concurrent systems by timed Petri nets*. Ph.d. thesis, M.I.T.

[Rousseau, 2010] Rousseau, J. (2010). *Contribution à la modélisation du trafic routier dans l'algèbre min-plus*. Thèse de doctorat, Université d'Angers.

[Su et al., 2009] Su, R., van Schuppen, J., and Rooda, J. (2009). The synthesis of time optimal supervisors by using heaps-of-pieces. *SE Technical Report No. 2009-8, Systems Engineering Group, Department of Mechanical Engineering, Eindhoven University of Techonology.*

[Wong-Toi and Hoffmann, 1995] Wong-Toi, H. and Hoffmann, G. (1995). The control of dense real-time discrete event systems. *Technical Report STA-CS-92-1411, Stanford University.*

Annexes

1. J. Komenda, S. Lahaye et J.-L. Boimond, *Supervisory Control of (max,+) Automata : a Behavioral Approach*, Discrete Event Dynamic Systems, vol. 19, num. 4, 2009

2. L. Houssin, S. Lahaye et J.-L. Boimond, *Just in Time Control of constrained (max,+)-Linear Systems*, Discrete Event Dynamic Systems, vol. 17, num. 2, 2007

3. L. Houssin, S. Lahaye et J.-L. Boimond, *Control of (max,+)-Linear Systems minimizing delays*, accepté à Discrete Event Dynamic Systems sous la réserve de modifications mineure en cours.