



HAL
open science

Validation de modèles qualitatifs de réseaux de régulation génique: une méthode basée sur des techniques de vérification formelle

Grégory Batt

► **To cite this version:**

Grégory Batt. Validation de modèles qualitatifs de réseaux de régulation génique: une méthode basée sur des techniques de vérification formelle. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 2006. Français. NNT: . tel-00012040

HAL Id: tel-00012040

<https://theses.hal.science/tel-00012040v1>

Submitted on 26 Mar 2006

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.

N° d'ordre :
N° attribué par la bibliothèque :

UNIVERSITÉ JOSEPH FOURIER - GRENOBLE I

THÈSE

pour l'obtention du titre de
Docteur de l'université Joseph Fourier - Grenoble I
spécialité : Informatique
au titre de l'école doctorale
Mathématiques, Sciences et technologies de l'information, Informatique

présentée et soutenue publiquement le 24/02/2006
par Grégory BATT

**Validation de modèles qualitatifs
de réseaux de régulation génique :
une méthode basée sur des
techniques de vérification formelle**

Composition du jury :

M. Nicolas HALBWACHS,	Président
M. John LYGEROS,	Rapporteur
M. Denis THIEFFRY,	Rapporteur
M. Gilles BERNOT,	Examineur
M. Johannes GEISELMANN,	Examineur
M. Hidde de JONG,	Directeur de thèse

Thèse préparée au sein de l'équipe HELIX à l'INRIA Rhône-Alpes

Remerciements

Je voudrais tout d'abord remercier tous les membres du jury, en particulier les rapporteurs, Denis Thieffry et John Lygeros, pour avoir accepté de juger la qualité de ce travail, et pour avoir accompli cette tâche difficile avec un grand soin.

Je voudrais également remercier quelques personnes qui ont particulièrement compté durant mes (longues!) années d'études. Je pense en particulier à Marc Sebbah, enseignant en classes préparatoires au Lycée Fermat (Toulouse), qui a su me transmettre sa passion pour la biologie et pour la recherche. Par ailleurs, mon parcours universitaire, d'abord en biologie puis en informatique, n'aurait pas été possible sans la confiance que m'ont accordé Ioan Negrutiu et Gilbert Brun, en biologie à l'ENS Lyon, et Luc Bougé et Nicolas Halbwachs, en informatique à l'ENS Lyon et à l'UJF.

J'ai passé quatre excellentes années au sein de l'équipe Helix de l'INRIA Rhône-Alpes, dirigée par François Rechenmann, puis par Alain Viari. L'atmosphère d'entraide et de sympathie qui y règne y est pour beaucoup. Merci à tous les membres d'Helix, passés et présents. Je remercie également l'INRIA et Patrice Garnier, PDG de Genostar, pour m'avoir permis de terminer le développement logiciel et le transfert technologique de mon travail dans les meilleures conditions, en fin de thèse.

Le caractère interdisciplinaire de mon travail m'a donné l'opportunité de collaborer avec des spécialistes de différents domaines scientifiques, notamment Jean-Luc Gouzé, Tewfik Sari, Giancarlo Ferrari-Trecate, Radu Mateescu, Hubert Garavel et Johannes Geiselmann. Grâce à leurs grandes qualités à la fois scientifiques et humaines, travailler avec eux fut toujours un réel plaisir. J'ai également eu la chance de travailler au quotidien dans la "GNA team", une équipe joyeuse, motivée et dynamique, composée entre autres de Michel Page, Delphine Ropers et Samuel Druhle. Les discussions, scientifiques ou non, avec toutes ces personnes ont été très enrichissantes.

Hidde de Jong a été pour moi un directeur de thèse exceptionnel durant toutes ces années. Tout en me laissant une grande liberté, ses lectures attentives et critiques de mon travail, et ses suggestions, ont largement contribué à (ré-)orienter mon travail dans des directions qui me semblent maintenant les bonnes. De plus, ses très grandes qualités humaines ont rendu notre travail toujours agréable et passionnant.

Je souhaite également remercier ma famille, celle de Céline, et mes amis proches, pour leur soutien inconditionnel. Enfin, ma plus grande gratitude revient à Céline, pour tout l'amour qu'elle m'a donné, et pour sa patience infinie, car c'est peu dire que j'ai souvent été peu disponible durant ces années. Un immense merci à elle.

Table des matières

I	Contexte	15
1	Introduction	17
1.1	Problématique	17
1.2	Approche	19
1.3	Contributions	20
1.4	Organisation du document	22
2	Analyse de systèmes hybrides par model checking	23
2.1	Systèmes hybrides et abstraction discrète	23
2.1.1	Systèmes hybrides	23
2.1.2	Exemple : système à deux réservoirs	24
2.1.3	Automates hybrides	25
2.1.4	Analyse par abstraction discrète	30
2.2	Model checking	35
2.2.1	Structures de Kripke	36
2.2.2	Logiques temporelles	36
2.2.3	Model checking et abstraction	41
II	Méthode	45
3	Modèles linéaires par morceaux (LPM) de réseaux de régulation génique	47
3.1	Modélisation par un système d'équations différentielles LPM	47
3.1.1	Exemple : réseau d'inhibition réciproque	47
3.1.2	Systèmes LPM avec fonctions en escalier	48
3.2	Solutions de systèmes d'équations différentielles LPM	49
4	Analyse symbolique par abstraction qualitative	53
4.1	Analyse mathématique des systèmes LPM et partition de l'espace des phases	53
4.1.1	Partition en domaines de mode	53
4.1.2	Analyse de la dynamique dans les domaines de mode	55
4.1.3	Partition en domaines de flot	58
4.1.4	Analyse de la dynamique dans les domaines de flot	61
4.2	Abstraction discrète	62
4.2.1	Système de transitions LPM	63
4.2.2	Système de transitions LPM qualitatif	64
4.3	Modèles qualitatifs	67
4.4	Calcul symbolique du système de transitions LPM qualitatif	69
4.4.1	Calcul de l'ensemble des domaines	69
4.4.2	Calcul des propriétés des domaines	69
4.4.3	Calcul des transitions entre domaines	71

5	Vérification par model checking	77
5.1	Formulation du système de transitions LPM qualitatif en structure de Kripke	77
5.2	Formulation de propriétés dynamiques en logique temporelle	78
5.2.1	Propriétés exprimables en CTL et en μ -calcul	79
5.2.2	Propriétés exprimables en μ -calcul seulement	81
5.3	Vérification de propriétés temporelles	82
5.3.1	Vérification de propriétés exprimées en CTL	83
5.3.2	Vérification de propriétés exprimées en μ -calcul propositionnel	85
III	Outil informatique	89
6	Genetic Network Analyzer 6.0	91
6.1	Utilisation de GNA 6.0	91
6.1.1	Généralités	91
6.1.2	Principales fonctionnalités	92
6.2	Implémentation de GNA 6.0	98
6.2.1	Architecture du logiciel	99
6.2.2	Aspects algorithmiques	100
IV	Applications	107
7	Analyse de l'initiation de la sporulation chez <i>B. subtilis</i>	109
7.1	Modélisation de l'initiation de la sporulation	109
7.1.1	Sporulation chez <i>B. subtilis</i>	109
7.1.2	Réseau de régulation génique contrôlant l'initiation de la sporulation	110
7.1.3	Modèle LPM du réseau contrôlant l'initiation de la sporulation	112
7.2	Analyse qualitative	114
7.2.1	Simulation de l'initiation de la sporulation	114
7.2.2	Apport de l'utilisation de la partition en domaines de flot	115
7.3	Validation du modèle	118
7.3.1	Validation par des données phénotypiques	118
7.3.2	Validation par des données d'expression génique	119
8	Analyse de la réponse au stress nutritionnel chez <i>E. coli</i>	125
8.1	Modélisation de la réponse au stress nutritionnel	125
8.1.1	Réponse au stress nutritionnel chez <i>E. coli</i>	125
8.1.2	Réseau de régulation génique contrôlant la réponse au manque de carbone	126
8.1.3	Modèle LPM du réseau contrôlant la réponse au manque de carbone	127
8.2	Analyse qualitative	128
8.2.1	Simulation de l'entrée en phase stationnaire	129
8.2.2	Simulation de la réentrée en phase exponentielle	129
8.3	Interprétation et validation du modèle	132
8.3.1	Interprétation biologique des résultats de simulation	132
8.3.2	Validation du modèle	134

V	Discussion et conclusions	139
9	Discussion	141
9.1	Analyse de modèles continus par abstraction discrète	141
9.1.1	Modèles continus	141
9.1.2	Utilisation d'abstraction discrète	142
9.1.3	Comparaison et évaluation	145
9.2	Analyse de modèles discrets par model checking	147
9.2.1	Modèles discrets	147
9.2.2	Utilisation du model checking	147
9.2.3	Comparaison et évaluation	149
10	Conclusions	153
10.1	Bilan	153
10.2	Perspectives	155
VI	Annexes	159
A	Preuves	161
A.1	Analyse de la dynamique des systèmes LPM	161
A.2	Abstraction discrète et modèles qualitatifs	162
A.3	Calcul symbolique du système de transitions LPM qualitatif	163
B	Exemples de fichiers	165
B.1	Modèles et conditions initiales pour GNA	165
B.2	Exports vers NuSMV	166
B.3	Exports vers CADP	168
	Bibliographie	169
	Résumé	187
	Abstract	188

Liste des figures

2.1	Exemple d'un système à deux réservoirs	25
2.2	Diagramme représentant la dynamique du système à deux réservoirs dans ses quatre états discrets	26
2.3	Automate hybride H modélisant le système à deux réservoirs	28
2.4	Représentation du flot et d'une trajectoire dans l'espace d'état continu de l'automate hybride H	29
2.5	Système de transitions H-TS associé à l'automate hybride H	29
2.6	Système de transitions quotient, H-QTS, de H-TS par la relation d'équivalence \sim	32
2.7	Graphe de transitions correspondant au système de transitions quotient H-QTS	32
2.8	H-QTS et H-TS ne sont pas bisimilaires	33
3.1	Exemple d'un réseau d'inhibition réciproque	48
4.1	Partition en domaines de mode de l'espace des phases correspondant au réseau d'inhibition réciproque	54
4.2	Analyse du système LPM (3.4)-(3.5) dans le domaine de mode régulier M^{11} : ensemble focal et direction du flot	56
4.3	Ensembles focaux des domaines de mode réguliers correspondant au système LPM (3.4)-(3.5)	56
4.4	Analyse du système LPM (3.4)-(3.5) dans le domaine de mode singulier M^4 : ensemble focal et direction du flot	57
4.5	Portrait de phase global correspondant au système LPM (3.4)-(3.5)	58
4.6	Partition des domaines de mode réguliers M^1 et M^{11}	59
4.7	Partition des domaines de mode singuliers M^4 et M^{12}	60
4.8	Partition en domaines de flot de l'espace des phases correspondant au réseau d'inhibition réciproque	61
4.9	(a) Exemple d'une trajectoire partant de $D^{1.1}$ et atteignant $D^{3.2}$ en passant par les points \mathbf{x}^0 , \mathbf{x}^1 , \mathbf{x}^2 et \mathbf{x}^3 aux instants respectifs 0, τ^1 , τ^2 et τ^3 . (b) Relation de satisfaction pour les points \mathbf{x}^0 , \mathbf{x}^1 , \mathbf{x}^2 et \mathbf{x}^3	64
4.10	Graphe de transitions d'états $G = (\mathcal{D}, \rightarrow_{\sim_{\Omega}})$ associé au système LPM (3.4)-(3.5) et description de deux exécutions $(D^{1.1}, D^{2.2}, D^{3.2}, D^{4.2}, D^{4.1})$ et $(D^{1.1}, D^{6.2}, D^{11.2}, D^{11.3})$	66
4.11	Calcul du pattern de signes de dérivées dans $D^{4.2}$: $D^{4.2} \models_{\sim_{\Omega}} Dsign = \{(0, -1)\}$	71
4.12	Calcul des transitions de type dim^+ entre domaines de flot	74
4.13	Calcul des transitions de type dim^- entre domaines de flot	75
5.1	Structure de Kripke Σ -KS associée au système de transitions LPM qualitatif Σ -QTS	78

5.2	Propriétés inexprimables en CTL. (a) Les propriétés $\phi_A = \mathbf{AFG}p$ et $\phi_B = \mathbf{AFAG}p$ ne sont pas équivalentes. (b) Les propriétés $\phi_C = \mathbf{EG}(p \rightarrow \mathbf{F}\neg p \wedge \neg p \rightarrow \mathbf{F}p)$ et $\phi_D = \mathbf{EG}(p \rightarrow \mathbf{EF}\neg p \wedge \neg p \rightarrow \mathbf{EF}p)$ ne sont pas équivalentes	82
5.3	Structure de Kripke filtrée associée au système de transitions LPM qualitatif Σ -QTS et obtenue à partir de la structure de Kripke Σ -KS	84
5.4	Traduction de la structure de Kripke Σ -KS représentée figure 5.1 en système de transitions étiquetées, $W(\Sigma\text{-KS})$, comme proposé dans [157]	86
6.1	Capture d'écran de GNA 6.0 montrant le modèle du réseau d'inhibition réciproque en cours d'édition	93
6.2	Capture d'écran de GNA 6.0 montrant des conditions initiales pour le modèle d'inhibition réciproque et faisant apparaître des informations sur le déroulement d'une simulation	95
6.3	Capture d'écran de GNA 6.0 montrant le résultat d'une simulation	97
6.4	Organisation des paquetages constituant GNA, et leurs relations avec le système d'exploitation	99
6.5	Diagramme de classes simplifié du kernel	100
6.6	Pseudocode donnant l'algorithme de calcul du graphe de transitions	101
6.7	Pseudocode donnant un algorithme pour la méthode <code>generateTransition()</code> de la classe <code>QState</code>	102
6.8	(a) Problème du calcul des voisins de $D^{12.1}$. (b) Représentation d'une région hyperrectangulaire « entourant » $D^{12.1}$ pouvant servir à définir un ensemble de successeurs potentiels de $D^{12.1}$	102
6.9	Ensemble de successeurs potentiels de $D^{12.1}$ obtenu par la méthode naïve	103
6.10	Représentation de deux régions hyperrectangulaires incluant $D^{12.1}$ et pouvant servir à définir deux ensembles de successeurs potentiels D' de $D^{12.1}$ tels que (a) $D^{12.1} \subseteq \partial D'$ et (b) $D' \subseteq \partial D^{12.1}$	103
6.11	Ensemble de successeurs potentiels de $D^{12.1}$ obtenu par la méthode optimisée	104
7.1	Conventions graphiques utilisées pour représenter les réseaux de régulation génique	111
7.2	Réseau de régulation génique contrôlant l'initiation de la sporulation chez <i>B. subtilis</i>	111
7.3	Description simplifiée du phosphorelais chez <i>B. subtilis</i>	112
7.4	Modèle du réseau génique contrôlant l'initiation de la sporulation chez <i>B. subtilis</i>	113
7.5	Graphe de transitions obtenu lors de la simulation de l'initiation de la sporulation chez <i>B. subtilis</i>	115
7.6	Evolution temporelle de la concentration de certaines protéines du réseau de l'initiation de la sporulation le long de deux chemins du graphe de transitions	116
7.7	Comparaison des prédictions obtenues par deux méthodes d'analyse qualitative de modèles LPM	117
7.8	Profil d'expression temporel de la protéine Spo0A	120
7.9	Profil d'expression temporel de la protéine σ^H	121
7.10	Profils d'expression temporel des protéines SinI et σ^F	122
7.11	Profil d'expression temporel de la protéine Hpr	123

8.1	Gènes, protéines et interactions clés, impliqués dans la réponse au manque de carbone chez <i>Escherichia coli</i>	126
8.2	(a) Contrôle de la formation du complexe AMPc-CRP. (b) Contrôle de la topologie de l'ADN	127
8.3	Modèle du réseau génique contrôlant la réponse au stress nutritionnel chez <i>E. coli</i>	128
8.4	Graphe de transitions obtenu lors de la simulation d'une transition de la phase exponentielle à la phase stationnaire chez <i>E. coli</i>	130
8.5	Evolution temporelle de la concentration des protéines et des ARN stables du réseau de la réponse au stress nutritionnel le long d'un chemin du graphe de transitions	131
8.6	Graphe de transitions obtenu lors de la simulation d'une transition de la phase stationnaire à la phase exponentielle chez <i>E. coli</i>	133
8.7	Profil d'expression temporel de <i>fis</i>	135
8.8	Activation du promoteur P1 de l'opéron <i>rrnB</i> par Fis	136
8.9	Evolution du niveau de superenroulement de l'ADN	137
9.1	Réseau intercellulaire de signalisation entre les protéines Delta et Notch. (b) Automate hybride $H_{\text{one_cell}}$ modélisant les interactions entre les protéines Delta et Notch dans une cellule.	143
9.2	(a) Représentation du flot dans l'espace d'état continu et de la partition initiale d'un automate hybride de dimension 2. (b) Représentation des transitions calculées entre états discrets. (c) Partition d'un état, ayant plus d'un successeur, en trois états, ayant chacun un unique successeur.	144
9.3	(a) Réseau de régulation biologique constitué d'un graphe de régulation biologique et d'un ensemble de paramètres. (b) Graphe de transition asynchrone. (c) Graphe de transitions synchrone, obtenu à partir de (b).	148

Liste des tables

2.1	Propriétés du système à deux réservoirs	26
5.1	Vérification de propriétés exprimées en CTL, portant sur la dynamique du réseau d'inhibition réciproque	83
5.2	Vérification de propriétés exprimées en CTL, avec contrainte d'équité, portant sur la dynamique du réseau d'inhibition réciproque	85
5.3	Traduction en μ -calcul modal de quelques formules exprimées en μ -calcul propositionnel, comme proposé par Mateescu [157].	87
5.4	Vérification de propriétés exprimées en μ -calcul propositionnel, portant sur la dynamique du réseau d'inhibition réciproque.	87
6.1	Comparaison de deux implémentations de GNA	104
7.1	Résultats de la simulation de la réponse de différentes souches de <i>B. subtilis</i> , sauvage et mutantes, placées dans des conditions environnementales défavorables	117
7.2	Phénotypes prédits obtenus par model checking et phénotypes observés, pour un certain nombre de souches de <i>B. subtilis</i> placées en conditions environnementales défavorables	119

Première partie

Contexte

1 Introduction

1.1 Problématique

Durant ces quinze dernières années ont été développées de nombreuses techniques de biologie moléculaire permettant d'analyser à grande échelle les systèmes biologiques. Il est, par exemple, désormais possible de séquencer des génomes entiers [128] ou de mesurer à l'échelle d'un organisme l'expression des gènes [148], la présence et la quantité des protéines [2, 175], ainsi que les interactions protéine/protéine ou protéine/ADN [144, 221].

Bien que souffrant encore de nombreuses limitations, les perspectives offertes par l'utilisation de ces méthodes ont renouvelé l'intérêt de la communauté scientifique à considérer l'étude des systèmes biologiques dans leur ensemble plutôt que l'étude de leurs éléments constitutifs pris isolément [135, 224]. *Comprendre*, et plus encore, *prédire* le fonctionnement et les propriétés émergentes d'un système à partir de la description, souvent partielle, de ses composants est un immense défi scientifique, aux potentielles retombées médicales et économiques considérables.

Les systèmes biologiques complexes sont assez naturellement représentés comme des réseaux d'interactions entre leurs différents composants [3]. Quatre types de réseaux d'interaction moléculaire ont été l'objet de la plupart des études récentes : les réseaux d'interaction protéine/protéine, les réseaux métaboliques, les réseaux de transduction de signaux, et les réseaux de régulation génique. Dans le travail présenté ici, nous nous intéressons aux réseaux de régulation génique, qui jouent un rôle fondamental dans le contrôle du fonctionnement et du développement des organismes vivants. On peut distinguer deux axes principaux de recherche pour l'étude de ces réseaux : l'*inférence* de grands réseaux à partir de données produites par des méthodes à haut débit, et l'*analyse*, en général par modélisation et simulation, des propriétés dynamiques de réseaux de taille souvent plus modeste. Ces deux approches sont complémentaires, puisque leurs objectifs sont respectivement d'élucider la structure des réseaux à partir de données, et de comprendre leur fonctionnement à partir de leur structure. Nous nous intéressons ici au problème de l'analyse des propriétés dynamiques des réseaux de régulation génique.

De nombreux mécanismes de régulation de l'expression des gènes ont été identifiés [146]. La régulation se fait pour la plupart des gènes principalement au niveau de la transcription, mais elle peut également avoir lieu durant l'épissage ou le transport de l'ARN messager (chez les eucaryotes), durant leur traduction, ou lors de la maturation des protéines. L'expression d'un gène est très souvent placée sous le contrôle d'autres gènes, via la synthèse de protéines régulatrices. C'est l'ensemble de ces interactions de régulation entre gènes qui forme un réseau de régulation génique. Pour la plupart des organismes, la structure de ces réseaux est encore largement méconnue. Toutefois, certains phénomènes étudiés chez des organismes modèles, tels que *Escherichia coli*, *Saccharomyces cerevisiae*, ou encore *Drosophila melanogaster*, sont contrôlés par des réseaux dont la structure est relativement bien caractérisée. Il est à noter que ces connaissances sont encore aujourd'hui plus le fruit de décennies d'études de génétique et de biologie moléculaire que de l'utilisation de méthodes à haut débit.

La connaissance de la structure des réseaux ne permet pas toujours d'en comprendre le fonctionnement. L'analyse du réseau contrôlant la réponse au stress nutritionnel chez

E. coli illustre bien ce problème. Ce réseau a fait l'objet de plusieurs décennies de recherches intensives [117, 194]. On peut raisonnablement penser que les régulateurs principaux impliqués dans ce processus, et leurs cibles, ont été identifiés. Mais malgré tous les travaux réalisés, il n'existe pas encore de vision globale du fonctionnement de ce réseau. Dans ce cas précis, et plus généralement pour de nombreux réseaux d'intérêt biologique, il est difficile de comprendre comment le comportement du réseau émerge des interactions entre ses composants. On peut trouver au moins deux raisons à cela. Premièrement, ces réseaux font en général intervenir un grand nombre de gènes, d'ARN messagers, de protéines et d'autres molécules. Deuxièmement, la présence de boucles de rétroaction rend leur dynamique particulièrement complexe. L'utilisation de modèles et le développement de méthodes et d'outils pour leur analyse est alors indispensable [112, 160].

Une grande diversité de formalismes mathématiques a été proposée pour modéliser les réseaux de régulation génique [65, 199]. On peut citer, entre autres, les équations différentielles ordinaires, partielles ou qualitatives, les équations stochastiques, les réseaux booléens et leurs généralisations, les réseaux de Petri, et les automates hybrides. Naturellement, chaque formalisme permet de représenter de façon détaillée, ou au contraire d'abstraire, différents aspects. On peut également comparer ces formalismes selon plusieurs critères, notamment selon s'ils sont discrets, continus ou hybrides (*i.e.* à la fois discrets et continus), déterministes ou stochastiques, qualitatifs ou quantitatifs.

Les approches les plus couramment utilisées sont certainement les approches quantitatives à base d'équations différentielles ou stochastiques. Ces approches permettent d'obtenir des prédictions numériques précises sur les propriétés dynamiques du système. Toutefois, des limites importantes à l'application de ces méthodes existent. Premièrement, les mécanismes des réactions ou des interactions biochimiques utilisés dans la formulation des modèles sont en général peu, voire pas, connus. Deuxièmement, les données quantitatives sur les paramètres du modèle et les concentrations moléculaires sont généralement absentes, même dans le cas de systèmes bien connus, ce qui limite fortement l'application des méthodes numériques standards. En dehors de ces limites, il est raisonnable de supposer que de nombreuses propriétés importantes du fonctionnement des êtres vivants ne dépend pas de valeurs numériques précises ou de mécanismes réactionnels précis. Ces propriétés qui sont invariantes pour une certaine gamme de valeurs de paramètres ou de mécanismes réactionnels, ou autrement dit, qui ne dépendent pas des détails quantitatifs, peuvent être qualifiées de *propriétés dynamiques qualitatives* [22]. Pour l'ensemble de ces raisons, nous nous focalisons dans ce travail sur des *approches qualitatives*, qui permettent d'inférer des propriétés qualitatives de la dynamique du système à partir des données actuellement disponibles, qui sont incomplètes et non quantitatives.

Ces approches donnent une description qualitative de l'ensemble des comportements possibles du système. Ainsi, les modèles peuvent être utilisés pour obtenir une meilleure compréhension du fonctionnement du système, ou pour obtenir une prédiction des comportements possibles du système placé dans différentes conditions environnementales ou dans différents contextes génétiques. Mais avant d'utiliser un modèle pour générer de nouvelles connaissances, il faut préalablement *tester sa validité*. Cette étape est d'autant plus importante que dans le contexte de la modélisation de réseaux de régulation génique, les systèmes modélisés sont complexes et encore imparfaitement connus. On dit qu'un modèle est *valide* si on ne peut déduire du modèle une contradiction avec l'ensemble des connaissances sur le système. En particulier, les prédictions obtenues à partir du modèle doivent être en accord avec les observations faites sur le système.

De nombreuses techniques expérimentales permettent d’obtenir des informations sur la dynamique des réseaux de régulation génique. On peut citer, par exemple, l’utilisation de puces à ADN [148] et de la RT-PCR temps réel [42], pour les méthodes à haut débit, et l’utilisation de gènes rapporteurs [183], pour les méthodes à faible débit. Les résultats obtenus prennent en général la forme de profils temporels représentant l’évolution du signal mesuré pour chaque molécule d’intérêt (ARN ou protéine). Toutefois, et ce malgré les efforts soutenus de la communauté pour améliorer ces techniques, il semble juste de dire que les résultats, même quantitatifs, obtenus par ces méthodes ne peuvent encore être interprétés avec confiance que de manière essentiellement qualitative. En particulier, le lien entre l’intensité du signal mesuré et les concentrations cellulaires des molécules d’intérêt n’est, en général, pas bien établi (bruit important, présence de nombreux biais lors de la préparation ou de la quantification du signal, etc.).

Etant donné que les réseaux de régulation génique sont grands, que leur dynamique est complexe, et que les données expérimentales disponibles sont en général peu fiables, tester la validité d’un modèle de réseau de régulation génique est une tâche difficile, et sujette à l’erreur. Dans ces conditions, réaliser ce test de façon intuitive et informelle peut entraîner le rejet à tort d’un modèle. Alors que de nombreuses méthodes ont été proposées pour la modélisation et la simulation de réseaux de régulation génique, le problème de la validation de ces modèles semble, jusqu’à récemment, avoir attiré moins d’attention. Dans ce travail, nous proposons de développer une méthode efficace pour tester si les prédictions obtenues à partir de modèles qualitatifs sont en accord avec les données expérimentales disponibles sur la dynamique du système.

1.2 Approche

Nous souhaitons tester la cohérence entre les prédictions obtenues par analyse qualitative des modèles et les données expérimentales sur la dynamique du système. Etant donné les limitations actuelles des méthodes expérimentales pour la quantification des concentrations cellulaires des ARN ou des protéines, nous choisissons d’utiliser seulement l’information qualitative correspondant aux changements dans le *sens de variation* du signal, et non pas son intensité. Autrement dit, nous supposons que les données expérimentales disponibles donnent des informations fiables sur les sens de variation des concentrations des protéines du réseau, mais pas nécessairement sur les valeurs de ces concentrations. Cette information est plus pauvre, mais son utilisation repose sur des hypothèses moins fortes. Ainsi, notre méthode ne repose pas sur l’hypothèse, peu réaliste, que le signal mesuré reflète exactement la concentration cellulaire de la molécule d’intérêt, mais simplement qu’ils varient dans le même sens.

Egalement, les modèles que nous considérons dans cette étude sont qualitatifs et hybrides, et sont définis par des équations différentielles linéaires par morceaux (LPM). Les modèles LPM ont été proposés par Glass et Kauffman dans les années 70 [99] puis généralisés par Mestl *et al.* [163]. Ces modèles négligent les détails des mécanismes de la régulation des gènes, en n’en conservant que les aspects essentiels. De plus, leur forme mathématique simple rend possible une analyse symbolique et qualitative de la dynamique des réseaux ainsi représentés [70]. Par la méthode proposée par de Jong et collègues [22, 70, 69], inspirée de la simulation qualitative [140], une représentation discrète de la dynamique du système dans l’espace des phases est obtenue sous la forme d’un graphe de transitions entre états. Les différents états discrets du graphe correspondent à des régions dans l’espace des phases

dans lesquelles la dynamique du système est qualitativement identique, et les transitions entre états correspondent à l'existence de trajectoires du système passant dans l'espace des phases d'une région à l'autre. Les différentes régions sont définies à partir d'une partition de l'espace des phases. Il a été montré que le graphe de transitions obtenu est invariant pour tout un ensemble de paramètres définis par des contraintes d'inégalité, qui peuvent être déduites de la littérature expérimentale. Des algorithmes dédiés ont été proposés pour le calcul efficace et symbolique du graphe de transitions, utilisant l'information qualitative des contraintes d'inégalité. Cette méthode a été implémentée dans un outil appelé Genetic Network Analyzer (GNA) [68], et appliquée à l'étude de plusieurs réseaux de régulation génique [67, 192, 222].

Deux problèmes se posent principalement à l'application de cette méthode pour la validation de modèles. Premièrement, les comportements prédits et observés ne sont pas directement comparables. Effectivement, avec la méthode présentée plus haut [70], on ne peut pas caractériser sans ambiguïté les signes des dérivées des solutions dans les différentes régions de l'espace des phases, et donc, on ne peut pas connaître les sens de variation des concentrations des protéines le long des comportements prédits. Nous proposons donc d'utiliser une méthode d'analyse qualitative différente, donnant une caractérisation plus fine des comportements prédits. La méthode que nous proposons dans cette thèse permettant notamment de connaître les signes des dérivées des solutions le long des comportements prédits. De plus, alors que la méthode proposée dans [70] est présentée dans un formalisme emprunté au raisonnement qualitatif, nous avons fait le choix de présenter la méthode proposée ici dans le cadre de la *théorie des systèmes hybrides*. Cette théorie propose des concepts clairement définis et des résultats théoriques puissants pour l'analyse des systèmes qui, comme ceux que nous souhaitons étudier, sont définis « par morceaux », et qui sont eux-même des cas particuliers de systèmes hybrides. Un concept central à notre analyse est l'*abstraction discrète*, qui permet d'établir précisément le lien existant entre la dynamique continue du système, définie par les équations différentielles LPM, et sa représentation discrète, le graphe de transitions.

Deuxièmement, les prédictions du comportement du système sont obtenues sous la forme d'un graphe de transitions entre états, dans lequel les chemins représentent les différents comportements possibles. Or la taille de ce graphe croît fortement avec la taille (nombre de gènes) et la complexité (nombre d'interactions) des réseaux. Tester manuellement si les comportements prédits satisfont les propriétés observées du système devient rapidement difficile et sujet à l'erreur. Il est donc nécessaire de disposer d'une méthode automatique et efficace pour réaliser cette tâche. Nous proposons d'utiliser le model checking. Le model checking est une collection de techniques de vérification formelle pour vérifier automatiquement si un système, représenté par un système de transitions discret, satisfait certaines propriétés temporelles. Des algorithmes et des outils ont été développés pour résoudre ce problème de façon particulièrement efficace. Là encore, nous nous placerons dans le cadre de la théorie des systèmes hybrides, dans lequel le problème de la vérification formelle de ces systèmes a été bien étudié, et qui offre des résultats qui nous permettront de garantir qu'un modèle ne peut être rejeté à tort.

1.3 Contributions

Plusieurs contributions ont été réalisées dans ce travail.

Premièrement, nous proposons une analyse reposant sur une représentation plus fine de

l'état du système que celle utilisée par de Jong *et al.* dans [70], qui est obtenue en utilisant une partition plus complexe de l'espace des phases. Cette représentation permet d'obtenir une meilleure description de la dynamique du système, et notamment de caractériser dans les différentes régions, les signes des dérivées des solutions. Il est par conséquent possible de connaître les sens de variation des concentrations des protéines le long des comportements prédits. Cette information est, comme vu précédemment, essentielle pour la validation de modèle à partir des données expérimentales actuellement disponibles. Toutefois, l'utilisation d'une partition plus complexe de l'espace des phases, en conjonction avec la présence de discontinuités dans le second membre des modèles LPM, a posé des problèmes non triviaux pour l'établissement de règles pour le calcul symbolique du graphe de transitions, qui demeurent facile à évaluer. Ces résultats ont fait l'objet de la publication [25].

Deuxièmement, nous proposons de combiner cette méthode avec des techniques de model checking, qui permettent de vérifier efficacement les propriétés des grands graphes que l'on obtient lors de l'analyse qualitative de modèles biologiquement intéressants. L'utilisation de model checking pour l'analyse de propriétés dynamiques de modèles de systèmes biologiques a également été récemment proposée dans [33, 85, 116, 190, 210]. Toutefois, ces travaux portent sur des modèles discrets. Dans ce travail, nous montrons que le model checking peut également être utilisé pour l'analyse de modèles plus conventionnels, continus ou hybrides, via l'utilisation d'abstractions discrètes. Ces résultats ont fait l'objet des publications [21, 23].

Troisièmement, la méthode présentée plus haut a été implémentée dans une nouvelle version de l'outil *Genetic Network Analyzer* (GNA 6.0). Cette implémentation a nécessité la réécriture d'une partie importante du moteur de calcul du graphe de transitions et l'ajout de fonctionnalités pour l'export des graphes vers plusieurs model checkers. GNA 6.0 est librement disponible pour la recherche académique¹. De plus, le logiciel fait actuellement l'objet d'un transfert technologique pour sa diffusion gratuite auprès d'utilisateurs académiques et sa commercialisation auprès d'utilisateurs non académiques par la société Genostar Technologies.

Et quatrièmement, cette méthode a été appliquée à la validation à partir de données expérimentales de deux modèles grands et complexes. Ces modèles, proposés par de Jong *et al.* [67] et par Ropers *et al.* [192], concernent respectivement l'initiation de la sporulation chez *B. subtilis* et la réponse au stress nutritionnel chez *E. coli*. Ces deux phénomènes, bien qu'ayant été beaucoup étudiés, ne sont encore que partiellement compris. Nous avons pu vérifier par model checking que, dans les deux cas, les prédictions obtenues étaient cohérentes avec la plupart des propriétés observées du système. Toutefois, quelques incohérences ont également été identifiées, suggérant que des révisions ou des extensions des modèles sont nécessaires, ou que des expériences complémentaires doivent être réalisées. Également, cette méthode a été utilisée pour mieux caractériser des prédictions nouvelles concernant l'existence d'oscillations dans la concentration de certaines protéines, consécutivement à une disponibilité soudaine en carbone, chez *E. coli*. En dehors d'une contribution à une meilleure compréhension du fonctionnement de ces systèmes, ces deux études de cas illustrent plus généralement que, par cette méthode, il est possible d'exprimer un large éventail de propriétés observables expérimentalement et de vérifier leur cohérence avec des prédictions obtenues pour des modèles complexes. Une partie de ces résultats a été publiée dans [24].

¹Contact : hidde.de-jong@inrialpes.fr.

1.4 Organisation du document

D'un point de vue méthodologique, ce travail consiste en la vérification de propriétés dynamiques d'une classe particulière de systèmes hybrides à l'aide de techniques de model checking. Nous commençons donc par faire l'état de l'art de ce problème, dans le second chapitre de cette première partie. Nous présentons quelques aspects importants de la théorie des systèmes hybrides et du model checking. Nous introduisons plus en détail la notion d'abstraction discrète, qui nous servira à établir une connexion entre ces deux théories.

La méthode de validation de modèle proposée dans la section précédente est développée dans la partie II. Dans cette partie, nous reprenons les principaux aspects de la méthode proposée par de Jong *et al.* dans [70], en les présentant dans un formalisme emprunté à la théorie des systèmes hybrides, et nous étendons cette méthode dans les deux directions proposées dans la section précédente, à savoir, réaliser une analyse plus fine de la dynamique, de façon à pouvoir prédire l'évolution au cours du temps du sens de variation des concentrations des protéines du réseau, et combiner cette méthode avec des techniques de model checking pour pouvoir tester automatiquement et efficacement si les prédictions satisfont les propriétés observées. La classe de modèles que l'on considère dans cette étude est présentée dans le premier chapitre de cette partie. Le second chapitre détaille la méthode proposée pour l'analyse symbolique et qualitative de ces modèles. Enfin, la combinaison de cette méthode avec des techniques de model checking, pour la vérification des propriétés dynamiques des modèles, est proposée dans le dernier chapitre de cette seconde partie.

La partie III présente l'implémentation de la méthode dans une nouvelle version de l'outil informatique Genetic Network Analyzer.

La partie IV concerne l'application de cette méthode à la validation de deux modèles LPM qualitatifs, à partir de données expérimentales. Nous présentons d'abord l'analyse d'un modèle de l'initiation de la sporulation chez *B. subtilis* proposé par de Jong *et al.* [67], puis l'analyse d'un modèle de la réponse au stress nutritionnel chez *E. coli* proposé par Ropers *et al.* [192].

La cinquième et dernière partie consiste en une discussion de la méthode dans le contexte d'autres travaux similaires, suivie d'une synthèse de ce travail et de diverses perspectives suggérées par cette étude.

2 Analyse de systèmes hybrides par model checking

Comme nous le verrons dans le chapitre 3, les réseaux de régulation génique sont assez naturellement modélisés comme des systèmes hybrides. Les systèmes hybrides sont des systèmes dynamiques dont le comportement fait intervenir à la fois une évolution continue et des phénomènes discrets. De nombreux travaux ont récemment été réalisés pour leur analyse, qui ont débouché sur une théorie des systèmes hybrides proposant un cadre conceptuel riche et des résultats puissants. Dans ce chapitre, nous exposons un ensemble de concepts et de résultats qui nous serviront par la suite. Plus spécifiquement, nous nous intéresserons aux travaux portant sur la vérification formelle de systèmes hybrides qui consiste à prouver qu'un système satisfait un ensemble de propriétés attendues, notamment dans notre cas, la cohérence avec les observations. Nous évoquerons plusieurs techniques développées pour la vérification formelle de systèmes hybrides, en nous focalisant sur l'une d'elle, qui combine des techniques d'abstraction discrète et de model checking.

2.1 Systèmes hybrides et abstraction discrète

2.1.1 Systèmes hybrides

D'une façon très générale, les systèmes dynamiques sont caractérisés par leur *état*, évoluant au cours du temps. L'ensemble des états possibles forme l'*espace d'état*. Une évolution du système correspond à une *trajectoire* dans l'espace d'état, également appelée une *exécution* du système.

Selon la nature de l'espace d'état, on distingue :

- les systèmes *continus*, lorsque l'espace d'état est continu, c'est-à-dire dans \mathbb{R}^n , pour un certain n , appelé *dimension* du système. Un espace d'état continu est également appelé *espace des phases* dans la terminologie des systèmes dynamiques. La dynamique du système est alors typiquement définie par un système d'équations différentielles (ou à différences) et une trajectoire correspond à une solution du système d'équations différentielles. Un corps en chute libre est un exemple très simple de système continu.
- les systèmes *discrets*, lorsque l'espace d'état est discret, c'est-à-dire correspond à un ensemble dénombrable (et souvent fini). La dynamique du système est alors typiquement définie par un automate. Une trajectoire correspond à une séquence de transitions entre états. Un interrupteur électrique, ayant deux positions (*on* et *off*), est un exemple classique de système discret.
- les systèmes *hybrides*, lorsque l'espace d'état comprend une composante discrète et une composante continue. L'espace d'état correspond alors souvent au produit d'un espace d'état continu et d'un espace d'état discret : $S = Q \times X$ où Q est dénombrable et $X \subseteq \mathbb{R}^n$. Un exemple classique de système hybride est un thermostat à deux états discrets, $q \in \{on, off\}$, contrôlant la température d'une pièce, $\mathbf{x} \in \mathbb{R}_{\geq 0}$. L'état du système est alors noté $s = (q, \mathbf{x})$.

En comparaison avec l'analyse des systèmes purement continus, étudiés dans la communauté des systèmes dynamiques et de l'automatique, et avec l'analyse des systèmes purement discrets, étudiés en informatique, le développement d'une théorie pour l'analyse des systèmes hybrides est relativement récent. Les premiers travaux datent du début des années 90 (voir notamment [5, 7, 103, 153, 171, 211]).

Pourtant, on rencontre naturellement ce type de système dans de nombreuses applications. On peut citer notamment le contrôle du fonctionnement de moteurs automobiles [19, 20], de processus biochimiques [82, 145] ou de production industrielle [177, 204], et le contrôle du trafic autoroutier [26, 124, 151] ou aérien [147, 217]. Une explication possible du développement relativement tardif de cette théorie est que l'analyse de ces systèmes est en général plus difficile que l'analyse des systèmes purement discrets ou purement continus, la dynamique discrète pouvant affecter la dynamique continue de façon non triviale, et réciproquement.

Afin d'illustrer l'analyse des systèmes hybrides, nous considérons dans la section suivante l'exemple d'un système hybride simple.

2.1.2 Exemple : système à deux réservoirs

Dans cet exemple, nous considérons un système composé de deux réservoirs cylindriques A et B, de section 1 m^2 . Le fond de chaque réservoir est percé d'un orifice circulaire de section s_i , $i \in \{a, b\}$. Également, chaque réservoir est rempli par un robinet muni d'une valve, notée V_i , $i \in \{a, b\}$. La valve possède deux états : $v_i \in \{on, off\}$, $i \in \{a, b\}$. Ouverte, le débit du robinet est constant et vaut I_i , $i \in \{a, b\}$. Fermée, le débit est nul. Le niveau d'eau dans chaque réservoir est noté x_i , $i \in \{a, b\}$. Nous supposons de plus que la valve du robinet d'un réservoir est commandée par un capteur détectant le niveau d'eau dans l'autre réservoir : v_a , l'état de la vanne V_a dépend de x_b et réciproquement. Plus précisément, nous supposons que le capteur du réservoir B est placé à une hauteur θ_b , et que si x_b dépasse θ_b , alors la vanne V_a est fermée, et qu'inversement si x_b est inférieur à θ_b , alors la vanne V_a est ouverte. La situation est symétrique pour x_a , θ_a et V_b . Le système est représenté figure 2.1. Bien que l'étude de cet exemple ne présente que peu d'intérêt en soi, il nous permettra d'illustrer aisément les concepts que nous introduisons dans ce chapitre et également de préparer l'analyse des réseaux de régulation génique que l'on propose dans les chapitres suivants, cet exemple simple partageant avec ces derniers plusieurs caractéristiques importantes.

Lorsque une vanne V_i est fermée, le réservoir correspondant, A ou B, se vide. La variation de la hauteur d'eau du réservoir obéit à l'équation

$$\dot{x}_i = -s_i \sqrt{2gx_i}, \quad i \in \{a, b\},$$

où g est l'accélération de la pesanteur. Lorsque la vanne est ouverte, la variation de la hauteur d'eau du réservoir obéit à l'équation

$$\dot{x}_i = I_i - s_i \sqrt{2gx_i}, \quad i \in \{a, b\}.$$

L'ouverture et la fermeture des vannes est supposée instantanée. Le système possède donc 4 états discrets, selon l'état, *on* ou *off*, des deux vannes, et peut être décrit par le diagramme donné dans la figure 2.2. Ainsi, lorsque les deux vannes sont ouvertes, c'est-à-dire lorsque $v_a = on$ et $v_b = on$, noté (on, on) , les variations de hauteurs d'eau suivent l'équation $\dot{x}_i = I_i - s_i \sqrt{2gx_i}$, $i \in \{a, b\}$. Également, les deux vannes ne peuvent être

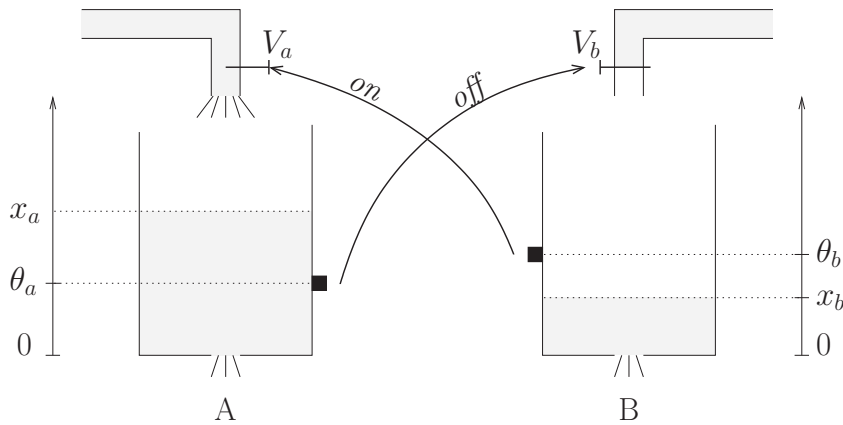


FIG. 2.1 – Représentation d'un système à deux réservoirs.

ouvertes que si les hauteurs d'eau dans les réservoirs sont toutes deux sous leur seuil : $0 \leq x_a \leq \theta_a$ et $0 \leq x_b \leq \theta_b$. Une information similaire est disponible pour chaque état discret (figure 2.2). Egalement, nous pouvons préciser sous quelles conditions le système change d'état discret. Si le système est dans un état discret (on, on) et si la hauteur d'eau dans le réservoir A, x_a , dépasse son seuil, θ_a , alors la vanne V_b est fermée, et le système passe dans l'état discret (on, off) . La situation est symétrique si x_b dépasse le seuil θ_b , le système passant alors dans l'état discret (off, on) . Notons toutefois l'existence d'un problème de modélisation. Que se passe-t-il si les deux hauteurs d'eau atteignent leur valeur seuil au même instant ? Si les deux valves se ferment, les niveaux d'eau dans les deux réservoirs redescendent immédiatement, ce qui cause alors la réouverture des deux valves, faisant réaugmenter les deux niveaux d'eau, causant la fermeture des deux valves, et ainsi de suite. En fait, ce problème est un cas particulier de problèmes qui se posent plus généralement lorsque que l'on utilise des équations différentielles à second membre discontinu. Nous proposerons une méthode pour traiter ce problème de façon mathématiquement correcte au chapitre 3. Ici, nous adoptons une solution simple, bien que peu satisfaisante, en posant arbitrairement que dans le cas où les deux hauteurs d'eau atteignent leur valeur seuil au même instant, seule la vanne V_b est fermée. Une solution *ad hoc* similaire est utilisée pour les autres états discrets afin d'éviter les problèmes causés par un changement simultané de l'état des deux valves (voir figure 2.2).

On peut se poser plusieurs questions sur le comportement de ce système. Nous en considérons deux. Premièrement, on peut se demander si, partant d'un état initial dans lequel les deux réservoirs sont vides (*i.e.* tels que $x_i = 0$, $i \in \{a, b\}$), le système peut atteindre un état dans lequel les deux réservoirs sont pleins (*i.e.* tels que $x_i > \theta_i$, $i \in \{a, b\}$). Et deuxièmement, est-ce qu'après un certain temps, on atteint toujours un état dans lequel un des deux réservoirs seulement est peu rempli (*i.e.* tels que $x_i < \theta_i$, $i \in \{a, b\}$) ?

Ces propriétés sont synthétisées dans la table 2.1. Nous allons voir dans la suite de ce chapitre comment répondre à ces questions.

2.1.3 Automates hybrides

De nombreuses représentations mathématiques des systèmes hybrides ont été proposées tant en automatique qu'en informatique [4, 8, 37, 39, 103, 153, 170, 172]. Dans cette section, nous présentons une classe de modèles fréquemment utilisé, permettant de représenter

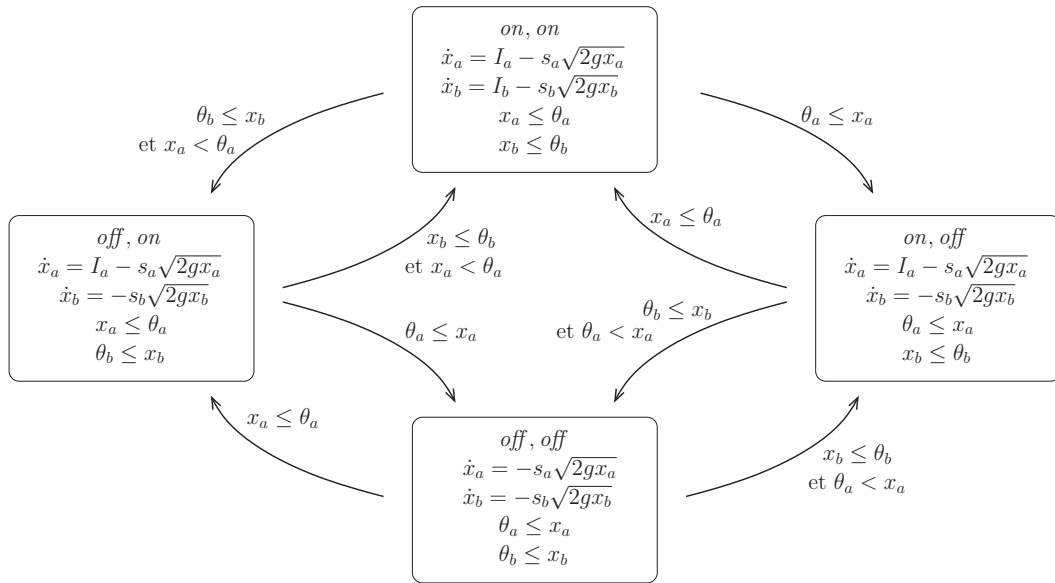


FIG. 2.2 – Diagramme représentant les quatre états discrets du système à deux réservoirs, leur dynamique et les transitions entre les états discrets.

-
1. Partant de l'état initial dans lequel les deux réservoirs sont vides ($x_a = 0$ et $x_b = 0$), le système peut atteindre un état dans lequel les deux réservoirs sont pleins ($x_a > \theta_a$ et $x_b > \theta_b$).
 2. Après un certain temps, le système atteint toujours un état dans lequel un des deux réservoirs seulement est peu rempli (soit $x_a \leq \theta_a$, soit $x_b \leq \theta_b$).
-

TAB. 2.1 – Deux propriétés à vérifier du système à deux réservoirs.

une grande diversité de dynamiques discrètes et continues : les *automates hybrides*. Les automates hybrides ont été proposés indépendamment par Alur et collègues [5] et par Nicollin et collègues [171]. Ils consistent en une généralisation des automates temporisés proposés par Alur et Dill [8]. Nous utilisons une syntaxe adaptée de Alur *et al.* [9].

Définition 1 (Automate hybride).

Un automate hybride est un tuple $H = (S, S_0, f, Inv, R, \Pi, \models)$ où :

- $S = Q \times X$ avec Q un ensemble fini, appelé espace d'état discret, et $X \subseteq \mathbb{R}^n$, appelé espace d'état continu,
- $S_0 \subseteq S$ est appelé ensemble d'états initiaux,
- $f : Q \rightarrow (X \rightarrow \mathbb{R}^n)$ est une application associant à tout état discret un champ de vecteur spécifiant le flot dans l'espace d'état continu.
- $Inv : Q \rightarrow 2^X$ est une application associant à chaque état discret une partie de X , appelée ensemble invariant,
- $R \subseteq S \times S$ est une relation décrivant les changements discontinus d'état,
- Π est un ensemble fini de propositions,
- $\models \subseteq S \times \Pi$ est une relation dite de satisfaction.

Nous supposons de plus que pour tout $q \in Q$, le champ de vecteur $f(q)$ est Lipschitz sur $Inv(q)$. Cette hypothèse garantit l'existence et l'unicité des solutions de l'équation différentielle $\dot{\mathbf{x}} = f(q)(\mathbf{x})$ pour toute condition initiale \mathbf{x}_0 dans $Inv(q)$. Cette solution est notée $\xi_q(\mathbf{x}_0, t)$, avec $t \geq 0$. Egalement remarquons que la définition utilisée ne permet de décrire que des systèmes hybrides autonomes. Enfin, l'ensemble Π est un ensemble de propositions servant à décrire l'état du système, et la relation de satisfaction définit quelles propriétés, π , sont vraies pour un état s du système, noté $s \models \pi$.

La sémantique d'un automate hybride est définie par un *système de transitions* associé. Nous empruntons les notations à Alur *et al.* [6].

Définition 2 (Système de transitions). Le système de transitions associé à l'automate hybride $H = (S, S_0, f, Inv, R, \Pi, \models)$ est le tuple H-TS = $(S, S_0, \rightarrow, \Pi, \models)$, où \rightarrow est une relation de transition entre états telle que $\rightarrow = \rightarrow_C \cup \rightarrow_D$, avec :

- $\rightarrow_C \subseteq S \times S$, décrivant des transitions dues à l'évolution continue :
pour tout $s = (q, \mathbf{x}) \in S$ et $s' = (q', \mathbf{x}') \in S$, $s \rightarrow_C s'$ si et seulement si

$$q = q' \text{ et il existe } t \geq 0 \text{ tel que } \xi_q(\mathbf{x}, t) = \mathbf{x}' \\ \text{et pour tout } t' \in [0, t], \xi_q(\mathbf{x}, t') \in Inv(q),$$

- $\rightarrow_D \subseteq S \times S$, décrivant des transitions discontinues :
pour tout $s = (q, \mathbf{x}) \in S$ et $s' = (q', \mathbf{x}') \in S$, $s \rightarrow_D s'$ si et seulement si

$$(s, s') \in R \text{ et } \mathbf{x}' \in Inv(q').$$

Il est important de remarquer que ce système de transitions contient un nombre infini d'états et de transitions entre états. Nous introduisons la notion d'*exécution* d'un système hybride, ou plus précisément, de son système de transitions associé. Une *exécution* est une séquence d'états, éventuellement infinie, telle qu'il existe une transition entre chaque paire d'états successifs. Plus précisément, (s_0, s_1, \dots) est une exécution de H si et seulement si

$$s_0 = (q_0, \mathbf{x}_0) \in S_0, \mathbf{x}_0 \in Inv(q_0) \text{ et pour tout } i \geq 0, s_i \rightarrow s_{i+1}.$$

Autrement dit, les exécutions sont composées d'une succession

- d'évolutions continues, durant lesquelles le système reste dans un même état discret, q , et satisfait l'équation différentielle $\dot{\mathbf{x}} = f(q)(\mathbf{x})$,
- et de transitions discontinues, instantanées, où l'état discret et/ou l'état continu peuvent changer.

Une exécution correspond à une trajectoire dans l'espace d'état S , ou encore à un comportement du système. Notons que, en toute généralité, pour certains états du système les deux types de transitions, évolution continue et transition discontinue, peuvent être possibles. Le système est alors non déterministe.

Afin d'illustrer ces concepts sur un exemple concret, définissons l'automate hybride H modélisant le système à deux réservoirs schématisé figure 2.1. Sa définition est donnée figure 2.3. H possède quatre états discrets, S , selon l'état, *on* ou *off*, des deux valves. Egalement, l'espace d'état continu X est $\mathbb{R}_{\geq 0}^2$, donnant les hauteurs d'eau dans les deux réservoirs. A chaque état discret, on associe une région de X , son invariant, qui traduit des contraintes physiques sur le système. Par exemple, l'invariant de l'état discret (*on, on*) traduit que les deux valves ne peuvent être ouvertes que si les niveaux d'eau sont inférieurs à leur seuil dans les deux réservoirs : $Inv(on, on) = \{\mathbf{x} \in X \mid x_a \leq \theta_a \text{ et } x_b \leq \theta_b\}$. Les propositions dans Π sont $\{inD^1, inD^2, inD^3, inD^4\}$, et la relation de satisfaction leur donne une interprétation simple : $(q, \mathbf{x}) \models inD^i$ si et seulement si $\mathbf{x} \in D^i$, où les ensembles D^i sont les invariants des états discrets. Enfin, les définitions de f et R ne font que reprendre les informations représentées figure 2.2.

$H = (S, S_0, f, Inv, R, \Pi, \models)$ où :

- $S = Q \times X$, avec $Q = \{(v_a, v_b) \mid v_a, v_b \in \{on, off\}\}$ et $X = \mathbb{R}_{\geq 0}^2$,
- $S_0 = \{(q, \mathbf{x}) \in S \mid \mathbf{x} \in Inv(q)\}$,
- $f : Q \rightarrow (X \rightarrow \mathbb{R}^2)$ telle que

$$\begin{aligned} f_a(q)(\mathbf{x}) &= f_a^{v_a}(x_a), \\ f_b(q)(\mathbf{x}) &= f_b^{v_b}(x_b), \end{aligned}$$

avec $q = (v_a, v_b)$, $f_i^{on}(x_i) = I_i - s_i \sqrt{2gx_i}$ et $f_i^{off}(x_i) = -s_i \sqrt{2gx_i}$, $i \in \{a, b\}$,

- $Inv : Q \rightarrow 2^X$ telle que

$$\begin{aligned} Inv(on, on) &= \{\mathbf{x} \in X \mid x_a \leq \theta_a \text{ et } x_b \leq \theta_b\}, \\ Inv(on, off) &= \{\mathbf{x} \in X \mid \theta_a \leq x_a \text{ et } x_b \leq \theta_b\}, \\ Inv(off, off) &= \{\mathbf{x} \in X \mid \theta_a \leq x_a \text{ et } \theta_b \leq x_b\}, \\ Inv(off, on) &= \{\mathbf{x} \in X \mid x_a \leq \theta_a \text{ et } \theta_b \leq x_b\}, \end{aligned}$$

- $R \subseteq S \times S$ est telle que $(s, s') \in R$ si et seulement si

$$\mathbf{x} = \mathbf{x}' \text{ et } \begin{cases} \mathbf{x} \in Inv(q'), & \text{si } v'_a = v_a, \\ \mathbf{x} \in Inv(q') \text{ et } x_a \neq \theta_a, & \text{sinon,} \end{cases}$$

avec $s = (q, \mathbf{x})$, $s' = (q', \mathbf{x}')$, $q = (v_a, v_b)$, et $q' = (v'_a, v'_b)$,

- $\Pi = \{inD^1, inD^2, inD^3, inD^4\}$,
- $\models \subseteq S \times \Pi$ est telle que

$$(q, \mathbf{x}) \models inD^i \text{ si et seulement si } \mathbf{x} \in D^i, \quad i \in \{1, \dots, 4\},$$

avec $D^1 = Inv(on, on)$, $D^2 = Inv(on, off)$, $D^3 = Inv(off, off)$, et $D^4 = Inv(off, on)$.

FIG. 2.3 – Automate hybride H modélisant le système à deux réservoirs représenté figures 2.1 et 2.2.

L'automate hybride H décrivant le système à deux réservoirs possède deux propriétés remarquables. Premièrement, au recouvrement des frontières près, les invariants forment

une partition de l'espace d'état continu (voir figure 2.4). Deuxièmement, la relation de transition R satisfait la propriété : si $(q, \mathbf{x}, q', \mathbf{x}') \in R$, alors $\mathbf{x} = \mathbf{x}'$. Autrement dit, il n'y a jamais de saut dans la valeur des variables continues : les solutions du système sont continues, bien que non nécessairement dérivables. Ces systèmes hybrides sont dits *continus par morceaux* [87, 182]. Les modèles de réseau de régulation générique que nous étudierons par la suite sont des cas particuliers de cette classe de systèmes hybrides.

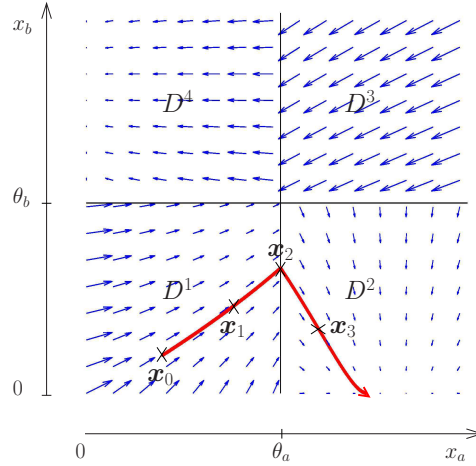


FIG. 2.4 – Représentation du flot et d'une trajectoire dans l'espace d'état continu de l'automate hybride H , défini figure 2.3. Les régions limitées par les plans de seuil, D^1, D^2, D^3 et D^4 , correspondent aux invariants des états discrets, $(on, on), (on, off), (off, off)$, et (off, on) , respectivement.

Une trajectoire décrivant l'évolution du système pendant un intervalle de temps fini est représentée figure 2.4. Plusieurs états du système peuvent être définis : $s_0 = (q_0, \mathbf{x}_0)$, $s_1 = (q_0, \mathbf{x}_1)$, $s_2 = (q_0, \mathbf{x}_2)$, $s_3 = (q_1, \mathbf{x}_2)$, et $s_4 = (q_1, \mathbf{x}_3)$, avec $q_0 = (on, on)$, $q_1 = (on, off)$, et $\mathbf{x}_0, \dots, \mathbf{x}_3$ comme représentés figure 2.4. Remarquons que deux états, s_2 et s_3 , sont associés au point \mathbf{x}_2 . Dans le système de transitions H-TS associé à l'automate hybride H , il existe, par exemple, une transition due à l'évolution continue de l'état s_0 à l'état s_1 et de s_1 à s_2 , et une transition discontinue de s_2 à s_3 , durant laquelle seul l'état discret change. La séquence $(s_0, s_1, s_2, s_3, s_4)$ est une exécution du système de transitions H-TS. Enfin, la seule proposition de Π satisfaite par l'état s_0 est inD^1 . Une description partielle du système de transitions infini H-TS est donnée figure 2.5.

H-TS = $(S, S_0, \rightarrow, \Pi, \models)$, où :

– $S = S_0 = \{s_0, s_1, s_2, s_3, s_4, \dots\}$,

– $\rightarrow = \{s_0 \rightarrow s_1, s_1 \rightarrow s_2, s_2 \rightarrow s_3, \dots\}$,

– $\Pi = \{inD^1, inD^2, inD^3, inD^4\}$,

– $\models = \{s_0 \models inD^1, s_1 \models inD^1, s_2 \models inD^1, s_3 \models inD^2, \dots\}$,

avec $s_0 = (q_0, \mathbf{x}_0)$, $s_1 = (q_0, \mathbf{x}_1)$, $s_2 = (q_0, \mathbf{x}_2)$, $s_3 = (q_1, \mathbf{x}_2)$, et $s_4 = (q_1, \mathbf{x}_3)$, en notant $q_0 = (on, on)$, $q_1 = (on, off)$, et $\mathbf{x}_0, \dots, \mathbf{x}_3$ représentés figure 2.4.

FIG. 2.5 – Représentation partielle du système de transitions infini H-TS associé à l'automate hybride H représenté figure 2.3.

Dans la section 2.1.1, nous avons énoncé deux propriétés sur le comportement du système que l'on souhaiterait vérifier (table 2.1). Quelles sont les méthodes développées permettant de répondre à ces questions, et plus généralement, permettant d'analyser les propriétés de la dynamique des systèmes hybrides ? Schématiquement, on peut classer les différentes méthodes en trois catégories [150]. Les approches *déductives* visent à construire une preuve de la propriété, souvent exprimée sous la forme d'un invariant, par induction [34, 38]. Bien que des assistants de preuve puissent être utilisés, l'application de ces techniques repose largement sur l'expertise de l'utilisateur. Il est également possible de vérifier certaines propriétés, d'atteignabilité notamment, en reformulant le problème original en un problème de *contrôle optimal* [152, 187]. La résolution analytique de ces problèmes est difficile, mais des solutions approchées peuvent être calculées numériquement. Enfin, on distingue les approches *algorithmiques*. Elles consistent soit à calculer de façon itérative l'espace atteignable (approches dites *directes*), soit à construire par abstraction discrète un système de transitions fini qui possède les mêmes propriétés que le système original, et que l'on peut analyser par des techniques classiques de model checking (approches dites *indirectes*). Les approches algorithmiques sont entièrement automatiques : étant donné une description du système hybride et de la propriété à vérifier, des outils permettent de tester automatiquement si la propriété est satisfaite par le système. La limite principale de l'applicabilité des approches algorithmiques est qu'elles ne donnent des résultats exacts que pour des classes restreintes de systèmes hybrides ou de propriétés. Pour de nombreuses autres classes, des résultats approchés (sur- ou sous-approximations) peuvent toutefois souvent être calculés de façon similaire.

Le travail présenté dans cette thèse repose sur l'utilisation d'une approche algorithmique indirecte particulièrement bien adaptée à la vérification de la classe de systèmes hybrides que nous utilisons pour modéliser des réseaux de régulation génique. Aussi, nous en donnons le principe dans le reste de cette section. Cette approche repose sur la notion d'*abstraction discrète*.

2.1.4 Analyse par abstraction discrète

Comme dit précédemment, l'objectif de cette méthode est de construire, à partir d'un système hybride, un système de transitions discret et fini possédant les mêmes propriétés que le système original, ou lorsque cela n'est pas possible, préservant un ensemble de propriétés d'intérêt (voir Alur *et al.* [9] pour une revue, et Alur *et al.* [6], Chutinan et Krogh [48] et Tiwari et Khanna [216] pour des travaux récents). La vérification des propriétés est alors réalisée sur le système discret par des techniques classiques de model checking (section 2.2). Plus précisément, cette méthode consiste à trouver une partition de l'espace d'état en régions dont tous les points peuvent être considérés comme équivalents par rapport aux propriétés à préserver. Une telle région correspond à un état dans le système de transitions discret. Dans cette section, nous formalisons ces concepts, en suivant l'approche proposée par Alur et collègues [9].

Étant donné l'ensemble de propositions Π , une relation d'équivalence $\sim \subseteq S \times S$ sur l'espace d'état est dite *préservant les propositions* si deux états équivalents satisfont le même ensemble de propositions. Formellement, \sim est telle que pour tout $s, s' \in S$ et pour tout $\pi \in \Pi$, si $s \sim s'$ et $s \models \pi$, alors $s' \models \pi$. Les classes d'équivalence d'une telle relation forment une *partition cohérente* (« consistent partition »), dans la terminologie de Chutinan et Krogh [48].

A partir d'une relation d'équivalence préservant les propositions, \sim , on définit naturellement le *système de transitions quotient* de H-TS par \sim .

Définition 3 (Système de transitions quotient). Le système de transitions quotient de H-TS $= (S, S_0, \rightarrow, \Pi, \models)$ par \sim est le tuple H-QTS $= (S/\sim, S_0/\sim, \rightarrow_\sim, \Pi, \models_\sim)$, où

- S/\sim et S_0/\sim sont les espaces quotients, *i.e.* les ensembles des classes d'équivalence, de S et S_0 ,
- $\rightarrow_\sim \subseteq S/\sim \times S/\sim$ est la relation de transition telle que pour tout $P, P' \in S/\sim$,

$$P \rightarrow_\sim P' \text{ si et seulement si il existe } s \in P \text{ et } s' \in P' \text{ tels que } s \rightarrow s',$$

- $\models_\sim \subseteq S/\sim \times \Pi$ est la relation de satisfaction telle que pour tout $P \in S/\sim$, et $\pi \in \Pi$,

$$P \models_\sim \pi \text{ si et seulement si il existe } s \in P \text{ tel que } s \models \pi$$

Autrement dit, les états du système de transitions quotient H-QTS sont les classes d'équivalence de la relation \sim . Egalement, il existe une transition entre deux états P et P' de H-QTS, si et seulement si il existe une transition entre un état de H-TS dans P et un état de H-TS dans P' . Enfin, un état P de H-QTS satisfait une propriété si et seulement si un état de H-TS inclus dans P satisfait cette propriété.

Les exécutions du système de transitions quotient H-QTS sont définies de façon analogue à celles de H-TS. Plus précisément, (P_0, P_1, \dots) est une exécution de H-QTS si et seulement si

$$P_0 \in S_0/\sim \text{ et pour tout } i \geq 0, P_i \rightarrow_\sim P_{i+1}.$$

A tout système de transitions quotient H-QTS, on associe le graphe de transitions $G = (X/\sim, \rightarrow_\sim)$, qui décrit simplement les relations d'atteignabilité entre classes d'équivalence.

On peut appliquer l'abstraction discrète à l'étude du système à deux réservoirs représenté par l'automate hybride H défini dans la figure 2.3. Soit \mathcal{D} l'ensemble des régions correspondantes aux invariants de H et représentées figure 2.4 : $\mathcal{D} = \{D^1, D^2, D^3, D^4\}$. Alors \sim est définie comme la relation d'équivalence sur l'ensemble des états S qui préserve l'état discret et l'appartenance à une région de \mathcal{D} : $s \sim s'$ si et seulement si $q = q'$ et il existe $D \in \mathcal{D}$ tel que $\mathbf{x}, \mathbf{x}' \in D$, avec $s = (q, \mathbf{x})$ et $s' = (q', \mathbf{x}')$. Nous pouvons vérifier que la relation \sim ainsi définie préserve les propositions de Π , qui concernent justement l'appartenance à une région de \mathcal{D} . L'ensemble des états possibles du système (*i.e.* des états $s = (q, \mathbf{x})$ tels que $\mathbf{x} \in \text{Inv}(q)$) appartient à quatre classes d'équivalence : $P^i = \{(q, \mathbf{x}) \in S \mid \text{Inv}(q) = D^i \text{ et } \mathbf{x} \in \text{Inv}(q)\}, i \in \{1, \dots, 4\}$

Le système de transitions quotient de H-TS par \sim , H-QTS, est défini dans la figure 2.6, et son graphe de transitions est représenté figure 2.7. H-QTS comprend 4 états, correspondant aux classes d'équivalence P^1, P^2, P^3 et P^4 . Par définition d'un système de transitions quotient, l'existence d'une transition dans H-TS implique l'existence d'une transition correspondante dans H-QTS. Ainsi, les transitions $s_1 \rightarrow s_2$ et $s_2 \rightarrow s_3$ dans H-TS (voir figure 2.5 pour la définition des états s_1, s_2 et s_3) impliquent respectivement l'existence des transitions $P^1 \rightarrow_\sim P^1$ et $P^1 \rightarrow_\sim P^2$ dans H-QTS. La transition $P^1 \rightarrow_\sim P^1$ est une boucle sur P^1 . Egalement, puisque l'état s_1 satisfait la proposition $\text{in}D^1$, l'état correspondant dans H-QTS, P^1 , satisfait également $\text{in}D^1$: $P^1 \models_\sim \text{in}D^1$.

L'étude de notre exemple simple montre clairement l'intérêt de l'abstraction discrète : passer d'une représentation continue du système à une représentation discrète, bien plus facile à analyser. De plus, il est possible de préciser le lien existant entre un système de transitions et son quotient à l'aide de la notion de *simulation*.

H-QTS = $(S/\sim, S_0/\sim, \rightarrow\sim, \Pi, \models\sim)$, où :

- $S/\sim = S_0/\sim = \{P^1, P^2, P^3, P^4\}$,
- $\rightarrow\sim = \{P^1 \rightarrow\sim P^1, P^1 \rightarrow\sim P^2, P^1 \rightarrow\sim P^4, P^2 \rightarrow\sim P^2, \dots, P^4 \rightarrow\sim P^4\}$,
- $\Pi = \{inD^1, inD^2, inD^3, inD^4\}$,
- $\models\sim = \{P^1 \models\sim inD^1, P^2 \models\sim inD^2, P^3 \models\sim inD^3, P^4 \models\sim inD^4\}$.

FIG. 2.6 – Représentation du système de transitions quotient H-QTS de H-TS par la relation d'équivalence \sim .

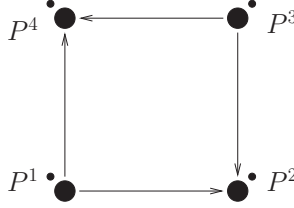


FIG. 2.7 – Représentation du graphe de transitions correspondant au système de transitions quotient H-QTS représenté figure 2.6. Les symboles \bullet représentent des boucles sur les états.

Définition 4 (Simulation [164]). Considérons deux systèmes de transitions définis sur un même ensemble de propositions Π , $M_1 = (S_1, S_{0_1}, \rightarrow_1, \Pi, \models_1)$ et $M_2 = (S_2, S_{0_2}, \rightarrow_2, \Pi, \models_2)$ et une relation préservant les propositions $\sim \subseteq S_1 \times S_2$. Alors, \sim est une *simulation* de M_1 vers M_2 si et seulement si, pour tout $s_1 \in S_1$ et $s_2 \in S_2$ tels que $s_1 \sim s_2$, on a

pour tout $s'_1 \in S_1$ tel que $s_1 \rightarrow_1 s'_1$, il existe $s'_2 \in S_2$ tel que $s_2 \rightarrow_2 s'_2$ et $s'_1 \sim s'_2$.

On dit que M_2 *simule* M_1 si il existe une simulation de M_1 vers M_2 telle que tout état initial de M_1 est équivalent à un état initial de M_2 . La terminologie s'explique par le fait que pour toute exécution de M_1 , il existe une exécution correspondante dans M_2 qui vérifie les mêmes propriétés. Par contre, une exécution de M_2 peut ne correspondre à aucune exécution de M_1 . Par définition, un système de transitions quotient simule le système de transitions original.

Proposition 1 (Système quotient et simulation). Considérons un système de transitions H-TS = $(S, S_0, \rightarrow, \Pi, \models)$ et une relation d'équivalence préservant les propositions $\sim \subseteq S \times S$. Alors le système de transitions quotient H-QTS = $(S/\sim, S_0/\sim, \rightarrow\sim, \Pi, \models\sim)$ de H-TS par \sim simule H-TS.

Une exécution présente dans le système quotient qui ne correspond à aucune exécution du système original est appelée artéfactuelle (« spurious behavior »). La présence possible d'exécutions artéfactuelles dans le système de transitions quotient suggère de définir une relation qui préserve des propriétés plus fortes entre les deux systèmes : une *bisimulation*.

Définition 5 (Bisimulation [176]). Soient deux systèmes de transitions définis sur un même ensemble de propositions Π , $M_1 = (S_1, S_{0_1}, \rightarrow_1, \Pi, \models_1)$ et $M_2 = (S_2, S_{0_2}, \rightarrow_2, \Pi, \models_2)$ et une relation préservant les propositions $\sim \subseteq S_1 \times S_2$. Alors, \sim est une *bisimulation* entre M_1 et M_2 si et seulement si, pour tout $s_1 \in S_1$ et $s_2 \in S_2$ tels que $s_1 \sim s_2$, on a

pour tout $s'_1 \in S_1$ tel que $s_1 \rightarrow_1 s'_1$, il existe $s'_2 \in S_2$ tel que $s_2 \rightarrow_2 s'_2$ et $s'_1 \sim s'_2$
pour tout $s'_2 \in S_2$ tel que $s_2 \rightarrow_2 s'_2$, il existe $s'_1 \in S_1$ tel que $s_1 \rightarrow_1 s'_1$ et $s'_2 \sim s'_1$.

Deux systèmes de transitions M_1 et M_2 sont *bisimilaires* si il existe une bisimulation entre M_1 et M_2 telle que tout état initial de M_1 est équivalent à un état initial de M_2 et réciproquement. La bisimulation est particulièrement intéressante, car deux systèmes bisimilaires satisfont exactement les mêmes propriétés, dans un sens que nous définirons précisément dans la section 2.2.3. Naturellement, un système de transitions quotient n'est pas nécessairement bisimilaire avec le système original.

Théorème 1 (Système quotient et bisimulation [4, 9, 35, 118]). Une relation d'équivalence préservant les propositions $\sim \subseteq S \times S$ est une bisimulation de H-TS $= (S, S_0, \rightarrow, \Pi, \models)$ si pour tout $s_1, s_2 \in S$ tels que $s_1 \sim s_2$, on a

pour tout $s'_1 \in S$ tel que $s_1 \rightarrow s'_1$, il existe $s'_2 \in S$ tel que $s_2 \rightarrow s'_2$ et $s'_1 \sim s'_2$.

Si \sim est une bisimulation, alors H-QTS est appelé un *quotient bisimilaire* de H-TS. Introduisons pour tout ensemble d'états $W \in 2^S$, l'ensemble des états prédécesseurs des états de W , $Pre(W) : Pre(W) = \{s \in S \mid \exists s' \in W, s \rightarrow s'\}$. Alors, une propriété caractéristique d'un quotient bisimilaire est que pour toute classe d'équivalence, $P \in S/\sim$, l'ensemble des états prédécesseurs des états de P , $Pre(P)$, est une union de classes d'équivalences [9].

La relation d'équivalence \sim proposée plus haut pour l'analyse par abstraction discrète du système à deux réservoirs n'est pas une bisimulation. On peut le vérifier en considérant les deux états $s_a = (q_a, \mathbf{x}_a)$ et $s_b = (q_b, \mathbf{x}_b)$, avec $q_a = q_b = (on, on)$, et \mathbf{x}_a et \mathbf{x}_b comme représentés figure 2.8. s_a et s_b sont équivalents et appartiennent à P^1 . Ces deux états n'ont chacun qu'un seul successeur, $s'_a = (q'_a, \mathbf{x}_a)$ et $s'_b = (q'_b, \mathbf{x}_b)$, respectivement, avec $q'_a = (on, off)$ et $q'_b = (off, on)$. Ces deux états successeurs ne sont pas équivalents, s'_a et s'_b appartenant respectivement à P^2 et à P^4 . Les conditions du théorème 1 ne sont donc pas satisfaites. Une autre caractérisation du fait que \sim n'est pas une bisimulation est donnée par le fait que s_a fait partie de $Pre(P^2)$, mais pas s_b , qui est pourtant dans la même classe d'équivalence que s_a .

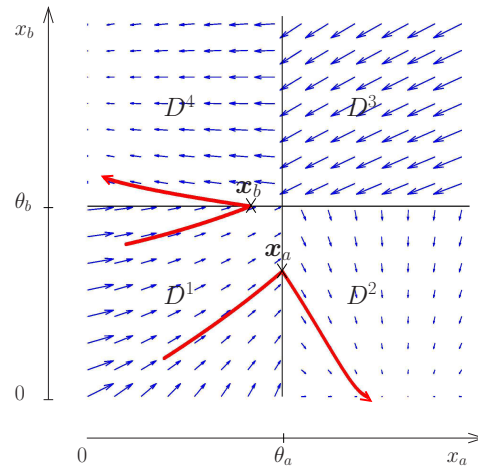


FIG. 2.8 – Représentation de deux trajectoires dans l'espace d'état continu de l'automate hybride H, illustrant que H-QTS et H-TS ne sont pas bisimilaires.

Bien que l'abstraction utilisée soit assez grossière (4 classes d'équivalence seulement) et que l'on obtient qu'un système de transitions H-QTS non bisimilaire au système de transitions original, H-TS, il est toutefois possible d'utiliser H-QTS pour répondre à la

première question posée à la section 2.1.1 (table 2.1). Cette propriété stipule que le système peut atteindre à partir de l'état initial $x_a = x_b = 0$ un état tel que $x_a > \theta_a$ et $x_b > \theta_b$. Ceci se traduit par l'existence d'une exécution de H-TS allant de l'état $s_0 = (q_0, \mathbf{x}_0)$ tel que $q_0 = (on, on)$ et $\mathbf{x}_0 = (0, 0)$ à un état $s = (q, \mathbf{x})$ tel que $q = (off, off)$, $x_a > \theta_a$ et $x_b > \theta_b$. Etant donné que H-QTS simule H-TS, l'existence d'une telle exécution implique l'existence d'une exécution de H-QTS allant de la classe d'équivalence de s_0 , P^1 , à la classe d'équivalence de s , P^3 . Or, cette exécution n'existe pas dans H-QTS (voir figure 2.7). La propriété est donc fautive. La seconde propriété est satisfaite si toutes les exécution de H-TS atteignent un état satisfaisant inD^2 ou inD^4 . Etant donné que H-QTS simule H-TS, la propriété 3 est satisfaite si toutes les exécution de H-QTS atteignent P^2 ou P^4 . Cette dernière propriété n'est pas vraie, car il existe des exécutions de H-QTS qui ne mènent ni à P^2 ni à P^4 : (P^1, P^1, P^1, \dots) et (P^3, P^3, P^3, \dots) . On ne peut donc pas conclure pour la seconde propriété. Pourtant, on peut montrer en utilisant des propriétés de convergence asymptotique des solutions que toutes les trajectoires du système atteignent P^2 ou P^4 après un temps fini. Les deux exécutions précédentes ne correspondent donc pas à des comportements réels du système, ce qui explique le résultat contre-intuitif obtenu. Nous verrons à la section 2.2.2 qu'il est possible d'ajouter des contraintes, dites d'*équité*, pour contourner ce problème.

Une bisimulation ou une simulation est dite finie si le nombre de classes d'équivalence est fini. Une (bi)simulation finie génère un quotient (bi)similaire fini, c'est-à-dire dont le nombre d'états et de transitions entre états est fini. La recherche de quotients bisimilaires finis pour différentes classes de systèmes hybrides a été l'objet d'intenses recherches (voir [9, 120] pour des revues). Effectivement, prouver l'existence d'un quotient bisimilaire fini pour une classe de systèmes hybrides permet d'obtenir immédiatement des résultats de décidabilité pour des problèmes d'atteignabilité, d'inclusion de langage et de vérification de cette classe de systèmes [9, 120]. Egalement, à partir de ces résultats, des outils ont été développés pour réaliser la vérification de certaines classes de systèmes hybrides, dont Uppaal [27], Kronos [64], et Hytech [119].

Toutefois, les classes connues de systèmes hybrides admettant des bisimulations finies possèdent en général soit une dynamique continue très simple, soit une dynamique discrète très simple. L'utilisation de quotients bisimilaires pour l'analyse de la plupart des systèmes hybrides d'intérêt pratique est donc limitée. De plus, il n'est pas toujours nécessaire de conserver toutes les propriétés du système pour vérifier les propriétés d'intérêt. Aussi, l'utilisation de systèmes de transitions quotients finis, mais non nécessairement bisimilaires, pour la vérification des systèmes hybrides a été l'objet de plusieurs travaux récents [6, 48, 56, 167, 216] utilisant la notion d'*abstraction de prédicats* (« predicate abstraction ») [106]. Les travaux présentés dans cette thèse, ainsi que ceux publiés précédemment par de Jong *et al.* [69], reposent également sur le calcul d'un système de transitions quotient fini.

Pour les systèmes hybrides modélisant des systèmes de complexité réaliste, comme les grands réseaux de régulation génique que nous souhaitons analyser, les systèmes de transitions quotients, similaires ou bisimilaires, contiennent souvent un grand nombre d'états et de transitions entre états. Il n'est alors plus possible d'analyser manuellement les systèmes de transitions discrets obtenus, comme nous l'avons fait sur l'exemple simple du système à deux réservoirs. Analyser de façon efficace les propriétés dynamiques de grands systèmes de transitions discrets est l'objectif même du model checking que nous présentons dans la section suivante.

2.2 Model checking

Le model checking a été développé initialement pour la vérification de systèmes réactifs, c'est-à-dire de systèmes en interaction permanente avec leur environnement, pour lesquels il est important de pouvoir garantir un fonctionnement correct (« safety-critical systems »). On trouve notamment parmi ces systèmes des protocoles de communication (par exemple, [51]) et des contrôleurs de processus industriels potentiellement dangereux ou de systèmes de transport (par exemple, [110, 113]). Son principe est simple. Tout d'abord le comportement du système est modélisé sous la forme d'un système de transitions discret. Puis, les propriétés attendues du comportement du système sont formalisées, souvent sous forme de formules de logique temporelle. On peut ensuite utiliser des outils, appelés model checkers, pour vérifier automatiquement si le (modèle du) système satisfait les propriétés attendues. Le model checker alors, soit confirme que le système satisfait la propriété, soit donne un contre-exemple, en général sous la forme d'une exécution qui viole la propriété. Pour de larges classes de propriétés, des algorithmes de vérification efficaces ont été proposés.

Les premiers algorithmes de vérification ont été proposés dans les années 80 [54, 188] et les premiers model checkers pouvaient analyser des systèmes contenant jusqu'à quelques dizaines de milliers d'états. Le problème principal est l'explosion du nombre d'états du système (« state space explosion »), souvent rencontré dans les applications réelles. Effectivement, les systèmes modélisés sont souvent des systèmes concurrents, où plusieurs processus évoluent en parallèle. Ceci conduit à une explosion combinatoire du nombre d'états du système (exponentiel en le nombre de processus). Pour lutter contre ce problème, tout un arsenal de techniques a été proposé. Citons notamment, l'utilisation de diagrammes de décision binaires (« binary decision diagrams ») [41, 63], et l'utilisation de solveurs de satisfaisabilité booléenne (« SAT solvers ») très efficaces [52], pour la vérification symbolique. Egaleme nt l'utilisation d'abstraction et l'exploitation de la modularité [58, 108, 149], la prise en compte d'événements indépendants (« partial-order reduction ») [100], et des symétries [79], ont contribué à rendre le processus de vérification encore plus efficace. Ces techniques ont permis d'identifier et de corriger des erreurs dans la conception de systèmes complexes (voir notamment [57, 92, 125], <http://www.inrialpes.fr/vasy/cadp/case-studies/> et <http://nusmv.irst.itc.it/> pour des exemples). Les model checkers sont actuellement utilisés dans l'industrie, en complément des méthodes de test.

L'objectif de ce chapitre n'est pas de passer en revue l'ensemble des travaux réalisés dans cette communauté (voir [162] pour une introduction et [30, 59] pour plus de détails). Ici, nous souhaitons plutôt introduire un certain nombre de notions qui nous seront utiles par la suite. Ainsi, bien que différents cadres formels existent, à la fois pour la définition du système (formalismes basés sur état, sur action, ou mixtes) et pour la spécification des propriétés du système (formalismes logique ou opérationnel), nous considérons dans cette section que le système est représenté par une structure de Kripke (formalisme basé sur état) et que les propriétés sont spécifiées par des formules de logique temporelle (formalisme logique).

De nombreuses logiques temporelles ont été proposées [78]. Dans ce travail, nous nous intéressons spécifiquement aux logiques dites *arborescentes*. Les logiques arborescentes permettent notamment d'exprimer l'*existence* d'une exécution du système vérifiant une propriété donnée. Les logiques arborescentes s'opposent sur ce point aux logiques dites *linéaires*, qui quantifient implicitement sur toutes les exécutions possibles. Comme nous le verrons à la section 5.2, il est essentiel de pouvoir exprimer qu'une propriété est vraie

pour une exécution du système, mais pas nécessairement pour toutes, dans le contexte de la validation de modèle à partir de données expérimentales.

Nous commençons par définir les structures de Kripke. Puis nous présentons deux logiques temporelles arborescentes, complémentaires sur les aspects de la facilité d'utilisation, de l'efficacité de la vérification, et de l'expressivité, appelées *Computation Tree Logic* (CTL) et le μ -calcul propositionnel. Enfin, nous décrivons le processus de vérification.

2.2.1 Structures de Kripke

Les structures de Kripke sont utilisées pour modéliser un système discret et son comportement. L'état du système est décrit à l'aide de propositions atomiques. On note AP , l'ensemble des propositions atomiques.

Définition 6 (Structure de Kripke). Une structure de Kripke sur AP est un tuple $KS = (S, S_0, \rightarrow_{KS}, \models_{KS})$ tel que

- S est un ensemble fini d'états,
- $S_0 \subseteq S$ est l'ensemble des états initiaux,
- $\rightarrow_{KS} \subseteq S \times S$ est une relation de transition totale (*i.e.* telle que tout état ait au moins un successeur),
- $\models_{KS} \subseteq S \times AP$ est une relation de satisfaction qui associe chaque état avec l'ensemble des propositions atomiques vraies dans cet état.

L'ensemble des états initiaux est souvent omis, les propriétés du système étant alors vraies ou fausses par rapport à un certain S_0 donné (par défaut, $S_0 = S$). La sémantique de la structure de Kripke est donnée par l'ensemble R de ses exécutions (ou « runs »). Une séquence infinie d'états $r = (s_0, s_1, \dots)$ est une exécution de KS si et seulement si

$$\text{pour tout } i \geq 0, \text{ on a } s_i \rightarrow_{KS} s_{i+1}.$$

Il est à noter qu'une exécution de KS est nécessairement infinie. Egalement, nous notons R_0 , l'ensemble des exécutions de KS telles que $s_0 \in S_0$. Enfin, $r(i)$ est le i ème état de l'exécution $r : r(i) = s_i, i \geq 0$, avec $r = (s_0, s_1, \dots)$.

Un système de transitions quotient correspond à une structure de Kripke, si le nombre de classes d'équivalence est fini et si la relation de transition est totale. Ainsi, le quotient bisimilaire H-QTS $= (S/\sim, S_0/\sim, \rightarrow_{\sim}, \Pi, \models_{\sim})$, associé à l'automate hybride H modélisant le système à deux réservoirs étudié dans la section précédente, est une structure de Kripke sur Π , que nous appellerons également H-KS. On vérifie facilement dans le graphe G représenté figure 2.7 que la relation de transition est bien totale. Nous rappelons qu'une boucle est une transition d'un état sur lui-même.

2.2.2 Logiques temporelles

Comme précisé plus haut, nous nous intéressons à deux logiques arborescentes : CTL et le μ -calcul propositionnel. Les formules de ces deux logiques sont interprétées sur les arbres d'exécutions de la structure de Kripke, obtenus en choisissant un état initial et en « dépliant » la structure en un arbre infini.

Computation Tree Logic

Une formule de CTL contient, en plus des propositions atomiques et des opérateurs logiques habituels, des *quantificateurs de chemins*, **A** et **E**, et des *opérateurs temporels*, tels que **X** et **U**. La syntaxe des formules de CTL est définie de façon inductive par :

Définition 7 (Syntaxe de CTL [55]).

- True et toutes les propositions atomiques $p \in AP$ sont des formules de CTL,
- si f_1 et f_2 sont des formules de CTL, alors $\neg f_1$, $f_1 \wedge f_2$, **AX** f_1 , **EX** f_1 , **A** $[f_1 \mathbf{U} f_2]$, et **E** $[f_1 \mathbf{U} f_2]$ le sont également.

X est l'opérateur « *next* ». Les formules **AX** f_1 , ou **EX** f_1 , signifient respectivement que la propriété f_1 est vraie pour tout, ou pour au moins un, successeur de l'état considéré. **U** est l'opérateur « *Until* ». Les formules **A** $[f_1 \mathbf{U} f_2]$ ou **E** $[f_1 \mathbf{U} f_2]$ signifient respectivement que pour toutes les exécutions, ou pour au moins une exécution, partant de l'état considéré, il y a un état futur satisfaisant f_2 , et tel que tous les prédécesseurs de ce dernier satisfont f_1 . Formellement, la sémantique des formules de CTL est définie inductivement par la fonction d'interprétation définie comme suit.

Définition 8 (Sémantique de CTL [55]).

$$\begin{aligned}
 \llbracket \text{True} \rrbracket &=_{\text{def}} S, \\
 \llbracket p \rrbracket &=_{\text{def}} \{s \in S \mid s \models_{\text{KS}} p\}, \\
 \llbracket \neg f_1 \rrbracket &=_{\text{def}} S \setminus \llbracket f_1 \rrbracket, \\
 \llbracket f_1 \wedge f_2 \rrbracket &=_{\text{def}} \llbracket f_1 \rrbracket \cap \llbracket f_2 \rrbracket, \\
 \llbracket \mathbf{A}X f_1 \rrbracket &=_{\text{def}} \{s \in S \mid \forall s' \in S. s \rightarrow_{\text{KS}} s' \implies s' \in \llbracket f_1 \rrbracket\}, \\
 \llbracket \mathbf{E}X f_1 \rrbracket &=_{\text{def}} \{s \in S \mid \exists s' \in S. s \rightarrow_{\text{KS}} s' \wedge s' \in \llbracket f_1 \rrbracket\}, \\
 \llbracket \mathbf{A}[f_1 \mathbf{U} f_2] \rrbracket &=_{\text{def}} \{s \in S \mid \forall r \in R. r(0) = s \implies \exists k \geq 0. r(k) \in \llbracket f_2 \rrbracket \wedge \forall k' < k. r(k') \in \llbracket f_1 \rrbracket\}, \\
 \llbracket \mathbf{E}[f_1 \mathbf{U} f_2] \rrbracket &=_{\text{def}} \{s \in S \mid \exists r \in R. r(0) = s \wedge \exists k \geq 0. r(k) \in \llbracket f_2 \rrbracket \wedge \forall k' < k. r(k') \in \llbracket f_1 \rrbracket\}.
 \end{aligned}$$

Pour une formule f , $\llbracket f \rrbracket$ représente l'ensemble des états du modèle satisfaisant f . Autrement dit, un état s satisfait une propriété f , noté $\text{KS}, s \models f$, si et seulement si $s \in \llbracket f \rrbracket$. La structure de Kripke KS satisfait la propriété f si tous ses états initiaux satisfont f : $\text{KS} \models f$ si et seulement si pour tout $s_0 \in S_0$, $\text{KS}, s_0 \models f$.

A partir des opérateurs précédemment définis, on peut définir d'autres opérateurs, également fréquemment utilisés, tels que :

- **AF** $f =_{\text{def}} \mathbf{A}[\text{True} \mathbf{U} f]$, qui exprime que, pour toutes les exécutions, f sera vraie dans un état futur (**F** pour “*Future*”),
- **EF** $f =_{\text{def}} \mathbf{E}[\text{True} \mathbf{U} f]$, qui exprime que, pour une exécution au moins, f sera vraie dans un état futur,
- **AG** $f =_{\text{def}} \neg \mathbf{E}F(\neg f)$, qui exprime que, pour toutes les exécutions, f sera vraie dans tous les états futurs (**G** pour “*Globally*”), et
- **EG** $f =_{\text{def}} \neg \mathbf{A}F(\neg f)$, qui exprime que, pour une exécution au moins, f sera vraie dans tous les états futur.

CTL peut être utilisé pour formaliser les propriétés du système à deux réservoirs, présentées dans la table 2.1. Nous avons vu à la section 2.1.4 qu'une condition nécessaire pour que la première propriété soit vraie est que dans H-QTS, P^3 soit atteignable à partir

de P^1 . Autrement dit, il faut tester si l'état satisfaisant inD^3 est atteignable à partir de P^1 . Ceci est alors équivalent à tester si

$$\text{H-KS}, P^1 \models \mathbf{EF} inD^3.$$

La seconde propriété est satisfaite si toutes les exécutions de H-QTS atteignent soit P^2 soit P^4 , c'est-à-dire des états satisfaisant soit inD^2 soit inD^4 . Cette dernière propriété se traduit en CTL par

$$\text{H-KS} \models \mathbf{AF} (inD^2 \vee inD^4).$$

Comme remarqué dans la section 2.1.4, cette propriété est fautive, à cause de la présence des exécutions (P^1, P^1, P^1, \dots) et (P^3, P^3, P^3, \dots) . Or, comme nous avons vu précédemment, le système ne peut rester toujours dans P^1 ou P^3 : il atteint P^2 ou P^4 en temps fini. Nous souhaitons donc ajouter comme contrainte que le système ne reste pas toujours dans P^1 ou P^3 . Ceci peut être réalisé à l'aide de contraintes dites d'équité.

La notion d'*équité* (« fairness ») a été introduite pour exprimer le fait que certains processus doivent progresser [13, 90]. Un exemple typique de propriété d'équité est « une propriété doit être satisfaite infiniment souvent », ou encore, « si une propriété est vraie infiniment souvent (par exemple, un processus demande une ressource), alors une autre le sera aussi (par exemple, il l'obtient) ». Ainsi, dans notre exemple précédent nous souhaitons exprimer que la propriété $inD^1 \vee inD^3$ ne doit pas être satisfaite infiniment souvent. Malheureusement, il n'est pas possible d'exprimer en CTL pur les propriétés d'équité. Toutefois, pour contourner ce problème, les algorithmes de vérification de CTL ont été étendus pour pouvoir prendre en compte un certain type de propriétés d'équité. Dans cette approche, des *contraintes d'équité* sont ajoutées au système (*i.e.* à la structure de Kripke), de telle sorte que les quantificateurs de chemins \mathbf{E} et \mathbf{A} ne quantifient plus que sur les *exécutions équitables*, c'est-à-dire les exécutions qui satisfont les contraintes d'équité infiniment souvent. Ainsi, si nous ajoutons la contrainte d'équité $\neg(inD^1 \vee inD^3)$ au modèle, la propriété $\mathbf{AF} (inD^2 \vee inD^4)$ est alors vraie, car toutes les exécutions équitables, c'est-à-dire toutes les exécutions sauf (P^1, P^1, P^1, \dots) et (P^3, P^3, P^3, \dots) , satisfont la propriété attendue. Notons que les propriétés testées ici sont triviales pour la structure de Kripke, très simple, considérée (voir figure 2.7). Toutefois, pour des systèmes plus complexes, vérifier manuellement ce type de propriété peut être difficile.

Des outils, notamment NuSMV, ont été développés pour la vérification automatique des propriétés exprimées en CTL. NuSMV est un model checker efficace intégrant des techniques basées sur l'utilisation de diagrammes de décision binaires pour la vérification symbolique de propriétés formulées en CTL et en LTL, et sur l'utilisation de solveurs de satisfaisabilité booléenne pour la vérification bornée de propriétés formulées en LTL [49, 50]. LTL est une autre logique temporelle, linéaire [78]. Mentionnons également que NuSMV peut prendre en compte des contraintes d'équités simples.

CTL est considérée comme une logique permettant d'exprimer des propriétés temporelles de façon relativement naturelle. De plus, des algorithmes très efficaces, de complexité linéaire en la taille de la formule et en la taille de la structure de Kripke, ont été proposés pour la vérification des formules de CTL [55]. Pour ces deux raisons, CTL est certainement une des logiques les plus utilisées. Toutefois, elle est relativement peu expressive, certaines classes importantes de propriétés ne pouvant être exprimées en CTL. Dans la section suivante, nous présentons une logique beaucoup plus expressive, le μ -calcul propositionnel.

μ -calcul propositionnel

Le μ -calcul est une logique modale qui utilise des opérateurs de point fixe [138]. Telle qu'introduite par Kozen [138], elle est interprétée sur des structures de Kripke dont les transitions sont étiquetées, appelées systèmes de transitions mixtes. Dans notre présentation, nous considérons le μ -calcul décrit dans [143] qui propose une interprétation des formules du μ -calcul sur les structures de Kripke classiques. Nous ne donnons ici qu'une description succincte du μ -calcul, le lecteur intéressé pouvant se référer à [138] et [143].

Les formules du μ -calcul sont construites à l'aide des opérateurs logiques classiques, des modalités de *possibilité* (\diamond) et de *nécessité* (\square) et d'opérateurs de *plus grand point fixe* (ν) et *plus petit point fixe* (μ). Nous définissons la syntaxe et la sémantique des formules du μ -calcul lorsqu'elles sont données sous forme normale positive, c'est-à-dire lorsque les négations ne sont appliquées qu'aux propositions atomiques et aux variables propositionnelles, ceci afin que la sémantique des opérateurs de point fixe puisse être correctement définie (critère de monotonie syntaxique de Kozen [138]). Formellement, étant donné l'ensemble AP de propositions atomiques et un ensemble PV de variables propositionnelles, les formules du μ -calcul sont définies inductivement comme suit.

Définition 9 (Syntaxe du μ -calcul).

- si $p \in AP$ et $X \in PV$, alors True, False, p , $\neg p$, et X sont des formules du μ -calcul,
- si f_1 et f_2 sont des formules du μ -calcul, alors $f_1 \wedge f_2$, $f_1 \vee f_2$, $\square f_1$, $\diamond f_1$, $\nu X.f_1$, et $\mu X.f_1$ le sont également.

L'interprétation des formules du μ -calcul se fait par rapport à une structure de Kripke et à un contexte propositionnel. Un contexte propositionnel $\rho : PV \rightarrow 2^S$ est une fonction partielle qui associe aux variables propositionnelles des ensembles d'états. La notation $\rho[U/X]$ représente le contexte propositionnel identique à ρ , sauf pour la variable X , qui prend pour valeur U . L'ensemble des états qui vérifient la propriété f dans le contexte ρ (qui doit initialiser toutes les variables propositionnelles libres dans f) est notée $\llbracket f \rrbracket \rho$. Dans la suite, σ représente soit μ soit ν . Formellement, la sémantique des formules du μ -calcul est définie inductivement comme suit :

Définition 10 (Sémantique du μ -calcul).

$$\begin{aligned}
 \llbracket \text{True} \rrbracket \rho &=_{\text{def}} S, \\
 \llbracket \text{False} \rrbracket \rho &=_{\text{def}} \emptyset, \\
 \llbracket p \rrbracket \rho &=_{\text{def}} \{s \in S \mid s \models_{\text{KS}} p\}, \\
 \llbracket \neg p \rrbracket \rho &=_{\text{def}} \{s \in S \mid s \not\models_{\text{KS}} p\}, \\
 \llbracket X \rrbracket \rho &=_{\text{def}} \rho(X), \\
 \llbracket f_1 \wedge f_2 \rrbracket \rho &=_{\text{def}} \llbracket f_1 \rrbracket \rho \cap \llbracket f_2 \rrbracket \rho, \\
 \llbracket f_1 \vee f_2 \rrbracket \rho &=_{\text{def}} \llbracket f_1 \rrbracket \rho \cup \llbracket f_2 \rrbracket \rho, \\
 \llbracket \square f_1 \rrbracket \rho &=_{\text{def}} \{s \in S \mid \forall s' \in S. s \rightarrow_{\text{KS}} s' \implies s' \in \llbracket f_1 \rrbracket \rho\}, \\
 \llbracket \diamond f_1 \rrbracket \rho &=_{\text{def}} \{s \in S \mid \exists s' \in S. s \rightarrow_{\text{KS}} s' \wedge s' \in \llbracket f_1 \rrbracket \rho\}, \\
 \llbracket \sigma X.f_1 \rrbracket \rho &=_{\text{def}} \sigma \Phi^\rho, \\
 &\quad \text{où } \Phi^\rho : 2^S \rightarrow 2^S \text{ est tel que } \Phi^\rho(U) = \llbracket f_1 \rrbracket \rho[U/X].
 \end{aligned}$$

D'après la définition ci-dessus, nous voyons que la modalité de possibilité, \diamond , sert à exprimer qu'une certaine propriété doit être vraie pour au moins un des successeurs, tandis

que la modalité de nécessité, \Box , est utilisée pour exprimer qu'une propriété doit être vraie pour tous les successeurs de l'état considéré. Elles sont respectivement équivalentes aux opérateurs CTL **EX** et **AX**. Egalement, les propriétés caractérisées par des plus petits points fixes correspondent en général à l'éventualité, tandis que celles caractérisées par des plus grands points fixes correspondent en général à l'invariance [78]. La relation de satisfaction pour le μ -calcul est définie de façon similaire à celle de CTL : $\text{KS}, s \models f$ si et seulement si $s \in \llbracket f \rrbracket$ et $\text{KS} \models f$ si et seulement si pour tout $s_0 \in S_0$, $\text{KS}, s_0 \models f$.

Pour illustrer ces notions, nous pouvons traduire en μ -calcul les propriétés portant sur le système à deux réservoirs exprimées en CTL à la section précédente. La première propriété s'exprime en CTL comme **EF** $\text{in}D^3$. Sa traduction en μ -calcul utilise la modalité de possibilité et l'opérateur de plus petit point fixe :

$$\text{H-KS}, P^1 \models \mu X.(\text{in}D^3 \vee \Diamond X).$$

La seconde propriété est une propriété d'inévitabilité : **AF** $(\text{in}D^2 \vee \text{in}D^4)$. Elle se traduit en μ -calcul par

$$\text{H-KS} \models \mu X.(\text{in}D^2 \vee \text{in}D^4 \vee \Box X),$$

où l'on peut noter l'emploi de la modalité de nécessité, \Box , pour exprimer la quantification sur toutes les exécutions. Enfin, comme l'équité ne peut être exprimée en CTL pur, nous avons dû rajouter une contrainte d'équité sur le système, pour que soient exclues certaines exécutions artéfactuelles (voir section précédente). Contrairement à CTL, il est possible d'exprimer directement l'équité en μ -calcul. La propriété suivante traduit que toutes les exécutions qui ne restent pas infiniment dans P^1 ou dans P^3 , atteignent nécessairement P^2 ou P^4 .

$$\text{H-KS} \models \nu X.((\text{in}D^1 \vee \text{in}D^3 \rightarrow \mu Y.(\text{in}D^2 \vee \text{in}D^4 \vee \Box Y)) \wedge \Box X).$$

Le μ -calcul est une logique très expressive [78]. Elle subsume CTL. En fait, elle subsume également CTL*, une extension de CTL que l'on obtient en relaxant la condition qui impose à tout opérateur temporel d'être immédiatement précédé d'un quantificateur de chemin [80]. Les quelques propriétés très simples portant sur le comportement du système à deux réservoirs que nous avons considérées peuvent être formulées en CTL, avec ajout éventuel de contraintes d'équité, et en μ -calcul. Toutefois nous verrons aux chapitres 5 et 8 des propriétés dont la formalisation en logique temporelle nécessitera toute l'expressivité du μ -calcul. Un premier inconvénient est que l'utilisation explicite des opérateurs de point fixe rend la formulation de propriétés difficile [36]. Toutefois, il est possible d'étendre le μ -calcul avec des expressions régulières, et ainsi, d'obtenir des descriptions plus concises et plus intuitives que celles obtenues avec des opérateurs de point fixe [159]. Un second inconvénient est que l'on ne connaît pas d'algorithme efficace (*i.e.* linéaire en la taille de la structure de Kripke et de la formule) pour la vérification des formules du μ -calcul général. Mentionnons toutefois que pour un fragment important du μ -calcul, le fragment dit d'alternance 1, des algorithmes efficaces existent [60], et que pour de nombreuses autres propriétés (formules de degré d'alternance pas trop élevé) la vérification reste possible, bien que moins efficace [10].

Des outils, notamment CADP, ont été développés pour la vérification de propriétés exprimées en μ -calcul [91]. CADP contient, entre autres, Evaluator, un model checker efficace pour la vérification à la volée de propriétés exprimées en μ -calcul régulier (*i.e.* avec expressions régulières) d'alternance 1, basé sur des techniques de résolution de système

d'équations booléennes [159], et un évaluateur pour XTL, un langage permettant d'exprimer des propriétés de μ -calcul régulier général [158]. Ces outils utilisent un formalisme basé sur les actions et non sur les états. Dans ce formalisme, le système est décrit par un système de transitions étiquetées et le μ -calcul utilisé est le μ -calcul modal. Notre problème étant posé dans un formalisme basé sur les états, il sera nécessaire de pouvoir passer d'un formalisme vers l'autre, ou plus précisément, de transposer le problème de la vérification d'une formule de μ -calcul propositionnel sur une structure de Kripke en un problème équivalent de vérification d'une formule de μ -calcul modal sur un système de transitions étiquetées. Le principe de cette transposition, qui utilise les résultats de Mateescu [157], est donné à la section 5.3.2.

2.2.3 Model checking et abstraction

Nous avons vu dans les sections précédentes, que le comportement d'un système hybride peut être représenté sous la forme d'un système de transitions discret, appelé structure de Kripke. Egalement, nous avons vu qu'il était possible d'exprimer des propriétés dynamiques du système en logique temporelle, dans notre cas, soit en CTL, soit en μ -calcul. Nous pouvons alors vérifier automatiquement et efficacement si le système satisfait les propriétés, à l'aide de model checkers. Les model checkers renvoient, en plus de la réponse Vrai/Faux, un *diagnostic*, qui prend souvent la forme d'une exécution qui satisfait (*témoin*) ou qui viole (*contre-exemple*) la propriété. L'analyse de ce diagnostic peut révéler une modélisation du système ou une spécification de la propriété incorrecte, ou dans le cas contraire, donner une indication du fonctionnement du système (*i.e.* pourquoi la propriété est effectivement vraie ou fausse).

Lors de l'analyse du système à deux réservoirs, le fait que H-KS, ou plus précisément H-QTS, soit une simulation de H-TS, nous a permis, par des raisonnements simples, de proposer des conditions nécessaires ou suffisantes pour que les propriétés portant sur le système hybride original H soient satisfaites ou invalidées, respectivement (voir notamment la section 2.1.4). Dans le reste de cette section, nous allons donner des critères systématiques précisant sous quelles conditions les conclusions obtenues pour le système abstrait sont également valables pour le système original.

Le cas le plus simple est lorsque le système abstrait est un quotient bisimilaire du système original. Effectivement, deux systèmes bisimilaires satisfont exactement le même ensemble de propriétés exprimables en μ -calcul (théorème 2).

Théorème 2 (Bisimulation et μ -calcul [149]). Soient M_1 et M_2 deux systèmes de transitions bisimilaires, alors pour toute formule ϕ du μ -calcul,

$$M_1 \models \phi \text{ si et seulement si } M_2 \models \phi.$$

Etant donné que le μ -calcul subsume CTL, la bisimulation préserve également les propriétés de CTL [40]. Ce théorème justifie le terme d'*approximations exactes* utilisé pour caractériser les quotients bisimilaires : leur nombre d'états est plus faible mais leurs propriétés restent les mêmes ! Malheureusement, comme dit plus haut, seuls des systèmes hybrides très simples admettent en général des bisimulations finies.

La simulation préserve des propriétés moins fortes. En fait, elle préserve les fragments dits *universels* des logiques. Les formules universelles sont des formules exprimant des propriétés qui doivent être vraies pour *toutes* les exécutions du système. L'intuition est

naturellement que si une propriété est vraie pour toutes les solutions du système abstrait alors elle l'est *a fortiori* pour toutes les exécutions du système original.

Formellement, la syntaxe des formules du μ -calcul universel, $\Box L\mu$, est donnée par la définition 9 (section précédente) dans laquelle on interdit l'utilisation de la modalité de possibilité, \Diamond , les formules étant toujours exprimées sous forme normale positive [149]. Autrement dit, la seule modalité autorisée est la modalité de nécessité, \Box , qui quantifie sur toutes les exécutions. On définit également le fragment universel de CTL, \forall CTL. Les formules de \forall CTL sont les formules de CTL qui, lorsqu'elles sont données sous forme normale positive, ne contiennent pas le quantificateur existentiel de chemins, **E** [108]. Autrement dit, ces formules ne contiennent que le quantificateur universel de chemins, **A**.

Théorème 3 (Simulation et μ -calcul universel [149]). Soient M_1 et M_2 deux systèmes de transitions tels que M_2 simule M_1 , alors pour toute formule ϕ du μ -calcul universel,

$$\text{Si } M_2 \models \phi, \text{ alors } M_1 \models \phi.$$

Etant donné que le μ -calcul universel subsume \forall CTL, la simulation préserve également les propriétés de \forall CTL [108]. Ce théorème traduit que si toutes les exécutions du système quotient vérifient une propriété, alors cette propriété est nécessairement vraie pour toutes les exécutions du système original, ou de façon équivalente, que si une propriété est vraie pour une exécution du système original, alors elle doit l'être également pour une exécution du système abstrait. Les quotients similaires sont des *approximations conservatives* du système original. A l'inverse, on ne peut rien conclure de propriétés satisfaites par une ou plusieurs exécutions du système abstrait puisqu'elles peuvent ne correspondre à aucune exécution du système original (exécutions artéfactuelles).

La seconde propriété testée sur notre système à deux réservoirs (table 2.1), **AF** ($inD^2 \vee inD^4$), est une propriété universelle. Cette propriété, testée avec des contraintes d'équité, est vraie pour le système abstrait, H-QTS. Elle est donc également vraie pour le système original. Par ailleurs, la négation de la première propriété testée, $\neg \mathbf{EF} inD^3$, équivalente à **AG** $\neg inD^3$, est également une propriété universelle qui est vraie pour le système abstrait. La conclusion obtenue pour le système abstrait est donc également valable pour le système original : la propriété 1 énoncée dans la table 2.1 n'est pas vraie.

En résumé, nous avons présenté dans ce chapitre une méthode pour l'analyse de la dynamique des systèmes hybrides. Cette méthode repose sur la notion d'abstraction discrète. Dans cette approche, nous définissons une relation d'équivalence sur les états qui préserve un certain nombre de propriétés simples (*i.e.* tous les états d'une même classe satisfont les mêmes propriétés). En réalisant le quotient du système hybride original par cette relation d'équivalence, on définit alors un système de transitions abstrait dans lequel un état discret représente une classe d'équivalence et les transitions entre états discrets représentent l'existence de comportements du système passant d'un état d'une classe d'équivalence à un autre état d'une autre classe d'équivalence. Si le système de transitions abstrait est fini, il peut être analysé à l'aide d'outils informatiques, des model checkers. Les propriétés de la dynamique du système sont alors formalisées en logique temporelle et les model checkers testent si le système de transitions abstrait satisfait les propriétés. Il est possible d'établir quelles classes de propriétés sont conservées par cette approche. Ainsi, si le système abstrait est un quotient bisimilaire du système original, toutes les propriétés exprimables en

μ -calcul ou en CTL sont préservées. Sinon, seules les propriétés universelles du μ -calcul ou de CTL sont garanties d'être préservées.

Notre motivation pour l'étude des systèmes hybrides est, comme nous le verrons dans le chapitre suivant, que les réseaux de régulation génique sont assez naturellement modélisés comme des systèmes hybrides, et plus précisément, comme des systèmes continus par morceaux. L'analyse par abstraction discrète de ces derniers est en général particulièrement bien adaptée. Toutefois, deux spécificités importantes sont à prendre en compte. Premièrement, très peu de données numériques pour la valeur des concentrations moléculaires et des paramètres sont disponibles pour ces systèmes. Deuxièmement, les réseaux de régulation génique biologiquement intéressants font en général intervenir un nombre élevé de gènes. Or l'analyse symbolique de systèmes hybrides de dimension élevée est notoirement difficile, et les méthodes généralistes existantes passent mal à l'échelle. Aussi, nous proposons de développer une méthode dédiée permettant une analyse symbolique et efficace de cette classe de systèmes. Cette méthode est exposée en détail dans la partie suivante.

Deuxième partie

Méthode

3 Modèles linéaires par morceaux (LPM) de réseaux de régulation génique

Dans cette partie, nous présentons une méthode d'analyse symbolique et qualitative de modèles linéaires par morceaux (LPM) de réseaux de régulation génique, et sa combinaison avec des techniques de model checking, permettant de tester efficacement si les prédictions obtenues par cette méthode sont en accord avec les données expérimentales disponibles sur le système.

Plus spécifiquement, dans ce chapitre, nous présentons les modèles LPM que nous utilisons pour modéliser les réseaux de régulation génique. Les modèles LPM sont des systèmes hybrides définis par un système d'équations différentielles où les variables continues représentent les concentrations des protéines et où des fonctions en escalier sont utilisées pour représenter les régulations géniques. L'utilisation de ces modèles pour l'analyse des réseaux de régulation génique a été proposée par Glass et Kauffman [99]. Gouzé et Sari ont proposé une extension de cette approche utilisant des inclusions différentielles pour résoudre certains problèmes mathématiques posés par ces modèles LPM [105]. de Jong et collègues ont proposé une approximation de cette extension, mieux adaptée à l'analyse algorithmique de ces systèmes [22, 69, 70]. Le travail présenté ici reprend ces travaux. L'aspect hybride, continu et discret, du fonctionnement de ces réseaux a également suscité l'intérêt de chercheurs dans les communautés des systèmes hybrides [29, 97] et de la biologie des systèmes [161].

Dans ce chapitre, nous introduisons également un exemple de réseau de régulation génique simple qui nous servira à illustrer tous les concepts introduits dans cette partie. La méthode d'analyse qualitative de modèles LPM proposée sera alors développée au chapitre suivant.

3.1 Modélisation par un système d'équations différentielles LPM

3.1.1 Exemple : réseau d'inhibition réciproque

Considérons le réseau d'inhibition réciproque représenté figure 3.1. Les gènes a et b codent les protéines A et B, qui contrôlent l'expression des gènes. Plus précisément, la protéine B inhibe l'expression du gène a lorsqu'elle est présente en concentration supérieure à un certain seuil, tandis que la protéine A inhibe l'expression du gène b lorsqu'elle est présente en concentration supérieure à un premier seuil, et l'expression de son propre gène, en concentration supérieure à un second seuil, plus élevé. La dégradation des protéines n'est pas régulée. Tout en étant simple, ce réseau d'inhibition réciproque est un composant élémentaire des réseaux réels plus complexes et nous permet d'analyser quelques phénomènes caractéristiques de la différenciation cellulaire [168, 186, 214].

Bien que les contextes soient très différents, nous pouvons noter certaines similarités

entre le fonctionnement de ce système et celui du système à deux réservoirs présenté dans la partie précédente (section 2.1.2). Dans les deux cas, le dépassement d'une certaine valeur seuil pour une variable cause l'arrêt d'un processus qui tend à faire augmenter l'autre variable. Le réseau de régulation génique représenté figure 3.1 et le système à deux réservoirs présenté dans la partie précédente sont deux exemples de systèmes d'inhibition réciproque.

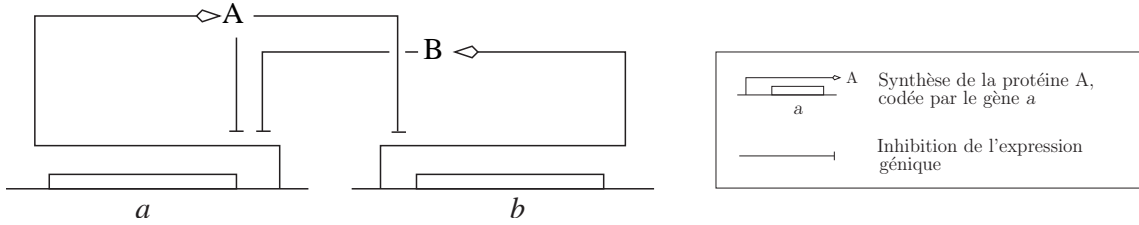


FIG. 3.1 – Exemple d'un réseau d'inhibition réciproque. Ce réseau simple est composé de deux gènes (a et b), codant chacun une protéine régulatrice (A et B).

3.1.2 Systèmes LPM avec fonctions en escalier

La dynamique des réseaux de régulation génique peut être représentée par une classe d'équations différentielles linéaires par morceaux dont la forme générale est la suivante [99, 163, 214] :

$$\dot{\mathbf{x}} = \mathbf{h}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) - \mathbf{g}(\mathbf{x}), \quad (3.1)$$

où $\mathbf{x} = (x_1, \dots, x_n)' \in \Omega$ est un vecteur de concentrations de protéines, $\mathbf{f} = (f_1, \dots, f_n)'$, $\mathbf{g} = \text{diag}(g_1, \dots, g_n)$, et $\Omega \subset \mathbb{R}_{\geq 0}^n$ est l'espace d'état, correspondant à une boîte de dimension n . Ω est également appelé espace des phases. Le taux de variation de chaque concentration protéique x_i , $i \in \{1, \dots, n\}$, est ainsi défini comme la différence entre le taux de synthèse $f_i(\mathbf{x})$ et le taux de dégradation $g_i(\mathbf{x}) x_i$ de la protéine. Ces modèles peuvent être facilement étendus afin de prendre en compte des variables d'entrées $\mathbf{u} = (u_1, \dots, u_m)'$, représentant la concentration de protéines et de petites molécules dont la synthèse et la dégradation sont régulées à l'extérieur du système [70].

La fonction $f_i : \Omega \rightarrow \mathbb{R}_{\geq 0}$ décrit comment le taux de synthèse de la protéine codée par le gène i dépend de la concentration \mathbf{x} des protéines de la cellule. Plus précisément, la fonction f_i est définie par

$$f_i(\mathbf{x}) = \sum_{l \in L_i} \kappa_i^l b_i^l(\mathbf{x}), \quad (3.2)$$

où $\kappa_i^l > 0$ est un paramètre de taux de synthèse, $b_i^l : \Omega \rightarrow \{0, 1\}$ une *fonction de régulation* constante par morceaux, et L_i un ensemble, éventuellement vide, d'indices de fonctions de régulation. La fonction g_i décrit la régulation de la dégradation de la protéine. Elle est définie de manière analogue à f_i , hormis le fait que g_i doit être strictement positive. De plus, afin de distinguer les paramètres de taux de dégradation des paramètres de taux de synthèse, on notera les premiers γ plutôt que κ . Une conséquence directe de ces définitions est que \mathbf{h} est une fonction *linéaire par morceaux*.

Une fonction de régulation décrit les conditions sous lesquelles la protéine codée par le gène i est synthétisée (ou dégradée) à un taux κ_i^l (ou $\gamma_i^l x_i$). Elle est définie à l'aide de fonctions en escalier et constitue l'équivalent arithmétique d'une fonction booléenne

exprimant la logique de la régulation génique [99, 163, 214]. Plus précisément, les conditions de synthèse ou de dégradation de la protéine sont exprimées à l'aide des fonctions en escalier s^+ , s^- :

$$s^+(x_j, \theta_j) = \begin{cases} 1, & \text{si } x_j > \theta_j, \\ 0, & \text{si } x_j < \theta_j, \end{cases} \quad \text{et } s^-(x_j, \theta_j) = 1 - s^+(x_j, \theta_j), \quad (3.3)$$

où x_j est une composante du vecteur de concentrations \mathbf{x} et θ_j un paramètre représentant un seuil de concentration. Trois vecteurs, $\boldsymbol{\kappa} = (\dots, \kappa_i^{k_i}, \dots)$, $\boldsymbol{\gamma} = (\dots, \gamma_i^{g_i}, \dots)$ et $\boldsymbol{\theta} = (\dots, \theta_i^{l_i}, \dots)$, représentent les valeurs des paramètres.

Les équations d'état modélisant le réseau d'inhibition réciproque de la figure 3.1 sont les suivantes :

$$\dot{x}_a = \kappa_a s^-(x_a, \theta_a^2) s^-(x_b, \theta_b) - \gamma_a x_a, \quad (3.4)$$

$$\dot{x}_b = \kappa_b s^-(x_a, \theta_a^1) - \gamma_b x_b, \quad (3.5)$$

où θ_a^1 et θ_a^2 représentent les concentrations seuils de la protéine A, avec $\theta_a^1 < \theta_a^2$, et θ_b , la concentration seuil de la protéine B. Afin de faciliter l'exposition de la méthode, nous supposons temporairement (jusqu'à la section 4.3) que les paramètres ont les valeurs suivantes : $\kappa_a = 18$, $\kappa_b = 7$, $\gamma_a = 2$, $\gamma_b = 1$, $\theta_a^1 = 4$, $\theta_a^2 = 8$, $\theta_b = 4$, $max_a = 12$ et $max_b = 8$, où max_a et max_b représentent les concentrations maximales des protéines A et B, respectivement.

L'équation (3.5) décrit le taux de variation de la concentration de la protéine B. Ainsi, la protéine B est produite (à un taux κ_b) si $s^-(x_a, \theta_a^1) = 1$, c'est-à-dire, si x_a , la concentration de la protéine A, est inférieure au seuil θ_a^1 . La protéine B est dégradée à un taux proportionnel à sa propre concentration ($\gamma_b > 0$ est une constante). L'équation (3.4) stipule que la protéine A est produite (à un taux κ_a), si la concentration de la protéine A est inférieure au seuil θ_a^2 ($s^-(x_a, \theta_a^2) = 1$) et x_b , la concentration de la protéine B est inférieure à θ_b ($s^-(x_b, \theta_b) = 1$). La protéine A est également dégradée à un taux proportionnel à sa propre concentration.

Il est à noter que nous aurions pu définir le système (3.4)-(3.5) sous la forme d'un automate hybride, de façon analogue à ce que nous avons fait pour le système à deux réservoirs dans la partie précédente. Toutefois, la notation proposée ici est plus compacte, puisqu'il n'est pas nécessaire d'énumérer pour tous les états discrets, la dynamique du système et les invariants. Egalement, comme nous allons le voir dans la section suivante, cette formulation facilite le traitement des problèmes liés à l'utilisation d'équations différentielles à second membre discontinu. Nous avons rencontré un cas particulier de ce problème dans la section 2.1.2 pour le système à deux réservoirs.

3.2 Solutions de systèmes d'équations différentielles LPM

Les équations différentielles présentées ci-dessus contiennent des fonctions en escalier $s^\pm(x_j, \theta_j)$ qui sont discontinues au point $x_j = \theta_j$. Ainsi, le second membre \mathbf{h} de (3.1) peut présenter des discontinuités dans l'union des hyperplans de seuil,

$$\Theta = \bigcup_{i \in \{1, \dots, n\}, l_i \in \{1, \dots, p_i\}} \{\mathbf{x} \in \Omega \mid x_i = \theta_i^{l_i}\},$$

où p_i représente le nombre de seuils de concentration de la protéine codée par le gène i . Ces discontinuités posent des problèmes non triviaux pour l'analyse du système. Afin d'apporter une solution à ces problèmes, nous utilisons une approche souvent utilisée en théorie du contrôle et proposée originellement par Filippov [87]. Elle consiste à étendre l'équation différentielle

$$\dot{\mathbf{x}} = \mathbf{h}(\mathbf{x}), \quad \mathbf{x} \in \Omega \setminus \Theta, \quad (3.6)$$

où \mathbf{h} est continue par morceaux sur Ω , et dont les points de discontinuités sont dans Θ , en l'inclusion différentielle

$$\dot{\mathbf{x}} \in K(\mathbf{x}), \quad \text{avec } K(\mathbf{x}) = \overline{\text{co}}(\{\lim_{\mathbf{y} \rightarrow \mathbf{x}, \mathbf{y} \notin \Theta} \mathbf{h}(\mathbf{y})\}), \quad \mathbf{x} \in \Omega, \quad (3.7)$$

où $\overline{\text{co}}(P)$ représente le plus petit ensemble convexe fermé contenant l'ensemble P , et $\{\lim_{\mathbf{y} \rightarrow \mathbf{x}, \mathbf{y} \notin \Theta} \mathbf{h}(\mathbf{y})\}$ représente l'ensemble de toutes les valeurs limites de $\mathbf{h}(\mathbf{y})$, pour $\mathbf{y} \notin \Theta$ et $\mathbf{y} \rightarrow \mathbf{x}$. L'utilisation de cette approche pour l'analyse de modèles linéaires par morceaux de réseaux de régulation génique a été récemment proposée par Gouzé et Sari [105].

En pratique, le plus petit ensemble convexe fermé $K(\mathbf{x})$ peut être difficile à calculer car $K(\mathbf{x})$ peut être un polyèdre complexe dans Ω . Aussi, nous utilisons une extension différente de l'équation différentielle (3.6),

$$\dot{\mathbf{x}} \in H(\mathbf{x}), \quad \text{avec } H(\mathbf{x}) = \overline{\text{rect}}(\{\lim_{\mathbf{y} \rightarrow \mathbf{x}, \mathbf{y} \notin \Theta} \mathbf{h}(\mathbf{y})\}), \quad \mathbf{x} \in \Omega, \quad (3.8)$$

où $\overline{\text{rect}}(P)$ représente le plus petit ensemble *hyperrectangulaire* contenant l'ensemble P [70]. Notons tout d'abord, que $H(\mathbf{x})$ est une surapproximation de $K(\mathbf{x})$, pour tout $\mathbf{x} \in \Omega$. Également, en utilisant $\overline{\text{rect}}$, il est possible d'exprimer $H(\mathbf{x})$ comme le produit de ses composantes, *i.e.* $H(\mathbf{x}) = H_1(\mathbf{x}) \times \dots \times H_n(\mathbf{x})$. Ainsi, l'inclusion différentielle (3.8) peut être réécrite comme le système d'inclusions différentielles $\dot{x}_i \in H_i(\mathbf{x})$, $i \in \{1, \dots, n\}$. Cette propriété est très importante pour le développement d'algorithmes efficaces, car elle permet d'analyser la dynamique du système dans chaque dimension séparément.

Formellement, nous définissons le système LPM Σ comme le triplet (Ω, Θ, H) , où Ω est l'espace des phases de dimension n , Θ est l'union des hyperplans de seuil et H la fonction définie en (3.8). Les solutions d'un système LPM sont les solutions de l'inclusion différentielle associée, (3.8).

Définition 11 (Solution d'une inclusion différentielle). Une solution de l'inclusion différentielle (3.8) sur un interval de temps I est une fonction absolument continue $\boldsymbol{\xi}(t)$ telle que

$$\dot{\boldsymbol{\xi}}(t) \in H(\boldsymbol{\xi}(t)), \quad \text{presque partout sur } I, \quad (3.9)$$

où « presque partout sur I » signifie partout sur I , excepté sur un ensemble de mesure nulle.

Cette définition de solution nous invite à considérer différents cas. Si à l'instant t une solution $\boldsymbol{\xi}$ est dans une région de l'espace des phases où \mathbf{h} est continue, c'est-à-dire, si $\boldsymbol{\xi}(t) \in \Omega \setminus \Theta$, alors $H(\boldsymbol{\xi}(t)) = \{\mathbf{h}(\boldsymbol{\xi}(t))\}$, et $\boldsymbol{\xi}$ est dérivable et satisfait $\dot{\boldsymbol{\xi}}(t) = \mathbf{h}(\boldsymbol{\xi}(t))$. Si dans une dimension i , $i \in \{1, \dots, n\}$, ξ_i atteint ou quitte une valeur seuil $\theta_i^{l_i}$ à l'instant t , la solution $\boldsymbol{\xi}$ peut ne pas être dérivable, et par conséquent, on peut ne pas avoir $\dot{\boldsymbol{\xi}}(t) \in H(\boldsymbol{\xi}(t))$. Enfin, si à l'instant t une solution glisse le long d'une surface de discontinuité, $\boldsymbol{\xi}$ est dérivable et $\dot{\boldsymbol{\xi}}(t) \in H(\boldsymbol{\xi}(t))$.

Pour tout $\mathbf{x}_0 \in \Omega$ et $\tau \in \mathbb{R}_{>0} \cup \{\infty\}$, $\Xi_\Sigma(\mathbf{x}_0, \tau)$ représente l'ensemble des solutions $\boldsymbol{\xi}$ du système LPM Σ pour la condition initiale $\boldsymbol{\xi}(0) = \mathbf{x}_0$ et sur l'intervalle de temps $[0, \tau]$, si τ est fini, ou $[0, \infty)$, sinon.¹ L'existence d'au moins une solution $\boldsymbol{\xi}$ sur un intervalle de temps $[0, \tau]$, $\tau > 0$, pour la condition initiale $\boldsymbol{\xi}(0) = \mathbf{x}_0$ est garantie, pour tout $\mathbf{x}_0 \in \Omega$ [87]. Toutefois, il n'y a en général pas de solution unique. On définit l'ensemble $\Xi_\Sigma = \bigcup_{\mathbf{x}_0 \in \Omega, \tau > 0} \Xi_\Sigma(\mathbf{x}_0, \tau)$ comme l'ensemble de toutes les solutions du système LPM Σ sur un intervalle de temps fini ou infini. Nous restreindrons notre analyse à l'ensemble des solutions dans Ξ_Σ qui atteignent ou quittent un hyperplan de seuil un nombre *fini* de fois.

Dans le chapitre suivant, nous développons une méthode bien adaptée à l'absence, ou à la rareté, des données quantitatives sur les paramètres du modèle et les concentrations moléculaires. Cette méthode permet l'analyse symbolique des modèles LPM, en exploitant leur forme mathématique simple et des informations qualitatives sur les valeurs des paramètres qui peuvent habituellement être déduites de la littérature expérimentale

¹Nous dirons par la suite, par un abus de terminologie, que $\boldsymbol{\xi}$ est une solution de Σ sur $[0, \tau]$, $\tau \in \mathbb{R}_{>0} \cup \{\infty\}$.

4 Analyse symbolique par abstraction qualitative

Dans ce chapitre, nous présentons une méthode d'analyse qualitative et symbolique de modèles LPM. Nous débutons par une analyse mathématique de la dynamique des systèmes LPM qui exploite le fait que ces systèmes se comportent localement de façon très simple. Plus précisément, nous proposons une partition de l'espace des phases en régions dans lesquelles la dynamique du système est qualitativement identique (section 4.1), puis nous utilisons une abstraction qualitative pour définir une représentation discrète de la dynamique, sous la forme d'un système de transitions LPM qualitatif (section 4.2). Nous montrons à la section 4.3 que ce système de transitions LPM qualitatif est invariant pour de larges gammes de valeurs de paramètres, définies par une classe de contraintes d'inégalité. Tous les modèles LPM correspondant satisfont donc les mêmes propriétés qualitatives. Cela nous conduit à l'introduction de la notion de modèle qualitatif. Enfin, nous montrons dans la section 4.4 comment calculer de façon symbolique le système de transitions LPM qualitatif correspondant à un modèle qualitatif.

La méthode présentée ici reprend, reformule, et étend la méthode proposée par de Jong *et al.* [70]. Le principe de l'analyse qualitative de modèles LPM est conservé. Toutefois, afin de réexploiter les travaux ayant très clairement établi un lien entre l'analyse des systèmes hybrides et le model checking, la méthode est reformulée dans un formalisme inspiré de la théorie des systèmes hybrides. Également, nous proposons un raffinement de la méthode proposée par de Jong *et al.* [70], qui permet d'obtenir une description plus précise de l'état du système, nécessaire à la comparaison non ambiguë entre les prédictions obtenues et les données expérimentales.

Les résultats présentés dans ce chapitre ont fait l'objet de la publication [25].

4.1 Analyse mathématique des systèmes LPM et partition de l'espace des phases

Dans cette section, nous introduisons une partition de l'espace d'état, également appelé espace des phases, en régions dans lesquelles le comportement du système est qualitativement identique. Cette partition sert de base à l'abstraction discrète qui sera introduite dans la section 4.2. Plus précisément, nous introduisons successivement deux partitions, la première (section 4.1.1) servant à définir la seconde (section 4.1.3), plus fine et préservant des propriétés plus fortes sur la dynamique du système.

4.1.1 Partition en domaines de mode

Considérons l'espace des phases de dimension n , $\Omega = \Omega_1 \times \dots \times \Omega_n$, où $\Omega_i = \{x_i \in \mathbb{R} \mid 0 \leq x_i \leq \max_i\}$ et \max_i représente la concentration maximale de la protéine codée par le gène i , $i \in \{1, \dots, n\}$. Les hyperplans de seuils $\{\mathbf{x} \in \Omega \mid x_i = \theta_i^{l_i}\}$, $i \in \{1, \dots, n\}$, $l_i \in \{1, \dots, p_i\}$, de dimension $n-1$, partitionnent l'espace des phases Ω en régions hyperrectangulaires. Étant donné que dans chacune de ces régions la régulation des gènes est

identique (section 4.1.2), nous dirons que chaque région correspond à un certain *mode de régulation*. Nous appellerons ces régions, des *domaines de mode*.

Définition 12 (Domaines de mode). L'ensemble des domaines de mode, \mathcal{M} , est l'ensemble des régions hyperrectangulaires $M = M_1 \times \dots \times M_n \subseteq \Omega$, telles que pour chaque $i \in \{1, \dots, n\}$, M_i satisfait l'une des équations suivantes :

$$\begin{aligned} M_i &= \{x_i \in \Omega_i \mid 0 \leq x_i < \theta_i^1\}, \\ M_i &= \{x_i \in \Omega_i \mid x_i = \theta_i^1\}, \\ &\dots \\ M_i &= \{x_i \in \Omega_i \mid x_i = \theta_i^{p_i}\}, \\ M_i &= \{x_i \in \Omega_i \mid \theta_i^{p_i} < x_i \leq \max x_i\}. \end{aligned}$$

La figure 4.1 montre la partition en domaines de mode de l'espace des phases de dimension 2 correspondant au réseau d'inhibition réciproque représenté figure 3.1. Nous distinguerons les domaines de mode dits *singuliers*, qui, comme M^7 et M^2 , sont localisés dans (l'intersection) des plans de seuils, des domaines de mode dits *réguliers*, qui, comme M^1 , ne le sont pas. Les ensembles des domaines de mode réguliers et singuliers sont notés respectivement \mathcal{M}_r et \mathcal{M}_s .

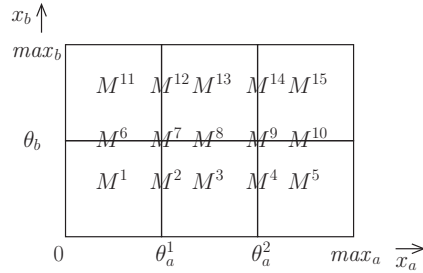


FIG. 4.1 – Partition en domaines de mode de l'espace des phases correspondant au réseau d'inhibition réciproque représenté figure 3.1.

Afin de préparer l'analyse des propriétés dynamiques des systèmes LPM, nous introduisons quelques concepts topologiques simple. A chaque région hyperrectangulaire $R \subseteq \Omega$ de dimension k , $k \in \{1, \dots, n\}$, est associé l'*hyperplan support* de R , $\text{supp}(R) \subseteq \Omega$, défini comme l'hyperplan de dimension k contenant R . La *frontière* de R dans $\text{supp}(R)$, ∂R , est définie comme l'ensemble des points $\mathbf{x} \in \text{supp}(R)$, tels que chaque boule $B(\mathbf{x}, \epsilon)$, de centre \mathbf{x} et de rayon $\epsilon > 0$ intersecte à la fois avec R et $\text{supp}(R) \setminus R$. Enfin, pour tout domaine de mode singulier M , $R(M)$ est défini comme l'ensemble des domaines de mode réguliers M' ayant M dans leur frontière.

$$R(M) = \{M' \in \mathcal{M}_r \mid M \subseteq \partial M'\}, \quad M \in \mathcal{M}_s.$$

Afin d'illustrer ces concepts sur quelques exemples, considérons à nouveau la partition de l'espace des phases représentée figure 4.1. On a $\text{supp}(M^1) = \Omega$, $\text{supp}(M^2) = \{\mathbf{x} \in \Omega \mid x_a = \theta_a^1\}$, et $\text{supp}(M^7) = \{\mathbf{x} \in \Omega \mid x_a = \theta_a^1 \text{ et } x_b = \theta_b\} = M^7$. Egalement, la frontière de M^2 dans $\text{supp}(M^2)$ correspond à M^7 , et celle de M^1 dans $\text{supp}(M^1)$, à l'union des domaines M^2 , M^6 et M^7 . Enfin, on a $R(M^2) = \{M^1, M^3\}$, $R(M^6) = \{M^1, M^{11}\}$ et $R(M^7) = \{M^1, M^3, M^{11}, M^{13}\}$.

4.1.2 Analyse de la dynamique dans les domaines de mode

Analyse dans les domaines réguliers

Dans chaque domaine de mode régulier, les fonctions b_i^l de l'équation (3.2) prennent une valeur constante, de telle sorte que les fonctions f_i et g_i se réduisent à une somme de constantes. Plus précisément, pour chaque domaine de mode régulier M , on a, pour tout $\mathbf{x} \in M$, $\mathbf{f}(\mathbf{x}) = \boldsymbol{\mu}^M$ et $\mathbf{g}(\mathbf{x}) = \boldsymbol{\nu}^M$, avec $\boldsymbol{\mu}^M = (\mu_1^M, \dots, \mu_n^M)' \geq \mathbf{0}$ et $\boldsymbol{\nu}^M = \text{diag}(\nu_1^M, \dots, \nu_n^M) > \mathbf{0}$. Nous pouvons alors proposer une écriture plus simple de $H(\mathbf{x}) = \overline{\text{rect}}(\{\lim_{\mathbf{y} \rightarrow \mathbf{x}, \mathbf{y} \notin \Theta} \mathbf{h}(\mathbf{y})\})$ définissant l'inclusion différentielle (3.8). Etant donné que la fonction \mathbf{h} est continue dans M , les valeurs des limites de \mathbf{h} en \mathbf{x} sont toutes égales à $\mathbf{h}(\mathbf{x})$. Par conséquent, l'ensemble $\{\lim_{\mathbf{y} \rightarrow \mathbf{x}, \mathbf{y} \notin \Theta} \mathbf{h}(\mathbf{y})\}$ est un singleton et vaut $\{\mathbf{h}(\mathbf{x})\}$ ou encore $\{\boldsymbol{\mu}^M - \boldsymbol{\nu}^M \mathbf{x}\}$. L'inclusion différentielle (3.8) se simplifie en :

$$\dot{\mathbf{x}} \in H(\mathbf{x}), \text{ avec } H(\mathbf{x}) = \{\boldsymbol{\mu}^M - \boldsymbol{\nu}^M \mathbf{x}\}, \mathbf{x} \in M, \quad M \in \mathcal{M}_r. \quad (4.1)$$

Naturellement, nous retrouvons dans ce cas régulier l'équation différentielle originale (3.1). A chaque domaine de mode régulier, M , est associé un *ensemble focal*, $\Psi(M)$, tel que toutes les solutions $\boldsymbol{\xi}$ dans M convergent monotoniquement vers $\Psi(M)$ [98, 184, 200].¹

Définition 13 (Ensemble focal : cas régulier). L'ensemble focal $\Psi(M)$ associé au domaine de mode régulier M est $\{\boldsymbol{\psi}(M)\}$, où $\boldsymbol{\psi}(M)$ est appelé le point focal, défini par $\boldsymbol{\psi}(M) = (\boldsymbol{\nu}^M)^{-1} \boldsymbol{\mu}^M$.

Nous faisons l'hypothèse générique que les ensembles focaux des domaines de mode réguliers ne sont pas situés dans Θ , l'union des plans de seuils.

Hypothèse 1. Pour tout $M \in \mathcal{M}_r$, $i \in \{1, \dots, n\}$ et $l_i \in \{1, \dots, p_i\}$, on a $\mu_i^M / \nu_i^M \neq \theta_i^{l_i}$.

Nous supposons de plus que pour tout $M \in \mathcal{M}_r$, et $i \in \{1, \dots, n\}$, on ait $\mu_i^M / \nu_i^M < \max_i$, ce qui garantit que toute solution $\boldsymbol{\xi} \in \Xi_\Sigma$ reste dans Ω . On peut montrer que si $\boldsymbol{\psi}(M) \in M$, alors toutes les solutions dans M approchent asymptotiquement $\boldsymbol{\psi}(M)$, qui est alors un point d'équilibre asymptotiquement stable du système. Dans le cas contraire, $\boldsymbol{\psi}(M) \notin M$, et toutes les solutions quitteront M après un certain temps [98, 184, 200].

Afin d'illustrer ces concepts, considérons le domaine de mode régulier M^{11} , représenté figure 4.2. En tout point \mathbf{x} de M^{11} , l'équation (3.5) se simplifie en $\dot{x}_b = \kappa_b - \gamma_b x_b$, puisque $s^-(x_a, \theta_a^1) = 1$. De même, l'équation (3.4) devient $\dot{x}_a = -\gamma_a x_a$. Alors, avec $\boldsymbol{\mu}^{M^{11}} = (0, \kappa_b)'$ et $\boldsymbol{\nu}^{M^{11}} = \text{diag}(\gamma_a, \gamma_b)$, l'ensemble focal associé à M^{11} , $\Psi(M^{11})$, est donné par $\{\boldsymbol{\psi}(M^{11})\}$, où $\boldsymbol{\psi}(M^{11}) = (0, \kappa_b / \gamma_b)'$. Avec les valeurs de paramètres données à la section 3.1.2, $\boldsymbol{\psi}(M^{11}) \in M^{11}$ (figure 4.2(a)), et toutes les solutions dans M^{11} convergent monotoniquement vers le point d'équilibre $\boldsymbol{\psi}(M^{11})$, comme illustré figure 4.2(b).

En général, des domaines de mode réguliers différents ont des ensembles focaux différents. Par exemple, $\Psi(M^1) = \{(\kappa_a / \gamma_a, \kappa_b / \gamma_b)'\}$. Les ensembles focaux des domaines de mode réguliers correspondant au système LPM (3.4)-(3.5) sont représentés figure 4.3.

Analyse dans les domaines singuliers

Dans un domaine de mode singulier, la dynamique du système, définie par $H(\mathbf{x}) = \overline{\text{rect}}(\{\lim_{\mathbf{y} \rightarrow \mathbf{x}, \mathbf{y} \notin \Theta} \mathbf{h}(\mathbf{y})\})$, dépend de la dynamique dans les domaines de $R(M)$. Effectivement, dans la définition de $H(\mathbf{x})$, $\mathbf{y} \rightarrow \mathbf{x}$ et $\mathbf{y} \notin \Theta$ implique que \mathbf{y} est situé dans un

¹Dans des travaux précédents, les ensembles focaux ont également été appelés *ensembles d'équilibres cibles* [67, 69, 70].

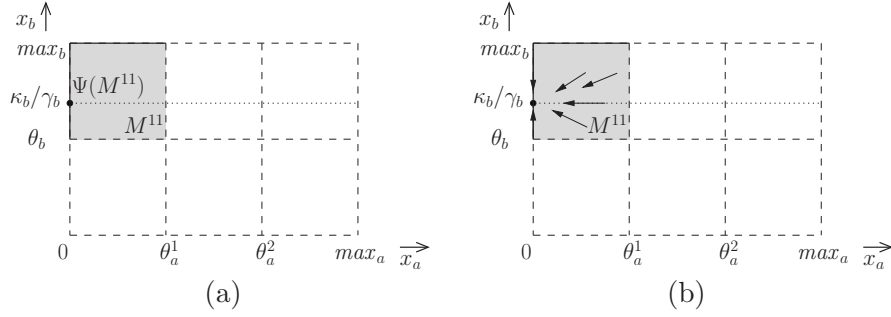


FIG. 4.2 – Analyse du système LPM (3.4)-(3.5) dans le domaine de mode régulier M^{11} : (a) ensemble focal et (b) direction du flot.

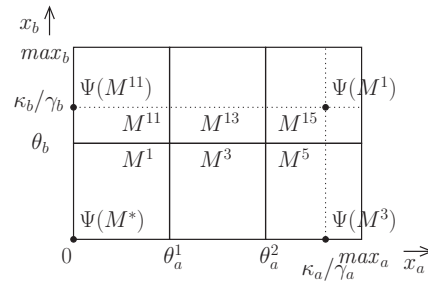


FIG. 4.3 – Ensembles focaux des domaines de mode réguliers correspondant au système LPM (3.4)-(3.5) (M^* représente M^5 , M^{13} et M^{15}).

domaine M' , régulier et ayant \mathbf{x} dans sa frontière, c'est-à-dire tel que $\mathbf{y} \in M' \in R(M)$. On déduit alors de la continuité de \mathbf{h} dans les domaines de mode réguliers que l'inclusion différentielle (3.8) se simplifie en

$$\dot{\mathbf{x}} \in H(\mathbf{x}), \text{ with } H(\mathbf{x}) = \overline{\text{rect}}(\{\boldsymbol{\mu}^{M'} - \boldsymbol{\nu}^{M'} \mathbf{x} \mid M' \in R(M)\}), \quad \mathbf{x} \in M, M \in \mathcal{M}_s. \quad (4.2)$$

Comme dans le cas régulier, on associe à chaque domaine de mode singulier un ensemble focal [70, 105].

Définition 14 (Ensemble focal : cas singulier). L'ensemble focal associé au domaine de mode singulier M est $\Psi(M) = \text{supp}(M) \cap \overline{\text{rect}}(\{\boldsymbol{\psi}(M') \mid M' \in R(M)\})$.

Nous distinguons deux cas, selon si $\Psi(M)$ est l'ensemble vide ou pas. Si $\Psi(M) = \emptyset$, toutes les solutions passant par M le traversent instantanément [70, 105]. M est alors appelé un domaine de mode *instantané*. Si $\Psi(M)$ n'est pas vide, alors certaines (mais pas nécessairement toutes) solutions restent dans M un certain temps, glissant le long d'hyperplans de seuil [70, 105]. Les domaines de mode non-instantanés, réguliers et singuliers, sont appelés *persistants*. L'ensemble des domaines de mode persistant est noté \mathcal{M}_{pers} : $\mathcal{M}_{pers} = \{M \in \mathcal{M} \mid \Psi(M) \neq \emptyset\}$. Notons que pour tout $M \in \mathcal{M}_{pers}$, $\Psi(M)$ est hyperrectangulaire et peut être écrit comme le produit de ses composantes : $\Psi(M) = \Psi_1(M) \times \dots \times \Psi_n(M)$.

Dans [70], il a également été prouvé que toutes les solutions dans un domaine de mode persistant M , convergent vers l'ensemble focal, $\Psi(M)$. Comme dans le cas régulier, la convergence vers $\Psi(M)$ des solutions dans M , $M \in \mathcal{M}_s$, est monotone, mais dans un sens faible : ξ_i converge monotoniquement vers $\Psi_i(M)$ pour tout $i \in \{1, \dots, n\}$ tel que $\xi_i(0) \notin \Psi_i(M)$ (voir [44] et lemme 1 dans l'annexe A). De plus, si $\Psi(M) \cap M$ n'est pas vide,

certaines solutions dans M atteignent ou approchent asymptotiquement $\Psi(M) \cap M$, qui est un ensemble de points d'équilibre du système. Dans le cas particulier où $\Psi(M) \subseteq M$, toutes les solutions dans M atteignent ou approchent asymptotiquement $\Psi(M)$. Si, au contraire, $\Psi(M) \cap M = \emptyset$, alors toutes les solutions quitteront M . Précisons que ces propriétés sont valables pour les *solutions dans M* , où par « solutions dans M », nous entendons « solutions restant dans M plus longtemps qu'un seul instant ».

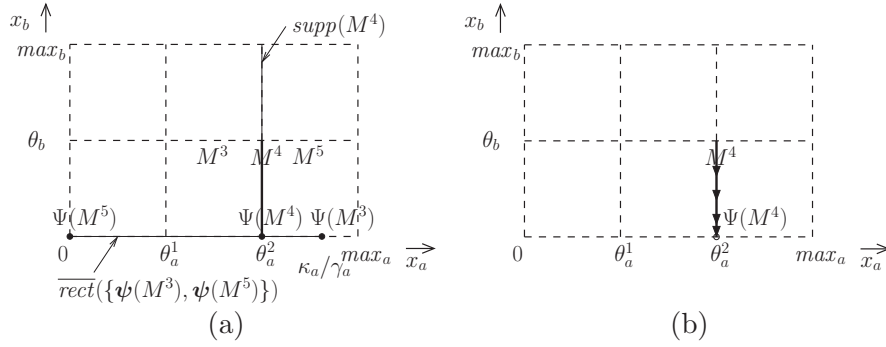


FIG. 4.4 – Analyse du système LPM (3.4)-(3.5) dans le domaine de mode singulier M^4 : (a) ensemble focal et (b) direction du flot.

Pour déterminer l'ensemble focal du domaine de mode singulier M^4 de notre exemple, nous appliquons la définition 14 (figure 4.4). L'hyperplan support de M^4 , $supp(M^4)$ est $\{\mathbf{x} \in \Omega \mid x_a = \theta_a^2\}$. Egalement, avec $R(M^4) = \{M^3, M^5\}$, l'ensemble $\overline{rect}(\{\psi(M') \mid M' \in R(M^4)\})$ est égal à $\{\mathbf{x} \in \Omega \mid 0 \leq x_a \leq \kappa_a/\gamma_a \text{ et } x_b = 0\}$. Donc, $\Psi(M^4) = supp(M^4) \cap \overline{rect}(\{\psi(M') \mid M' \in R(M^4)\})$ est égal à $\{(\theta_a^2, 0)\}$ (figure 4.4(a)). Puisque $\Psi(M^4)$ n'est pas vide, M^4 est persistant, et les solutions dans M^4 , glissant le long de l'hyperplan de seuil θ_a^2 , convergent vers $\Psi(M^4)$ (figure 4.4(b)). De plus, puisque $\Psi(M^4) \subseteq M^4$, toutes les solutions dans M^4 atteignent $\Psi(M^4)$, pour $t \rightarrow \infty$, qui est un point d'équilibre. Une analyse similaire peut être faite pour le domaine de mode singulier M^{12} . Avec $R(M^{12}) = \{M^{11}, M^{13}\}$, nous obtenons $\overline{rect}(\{\psi(M') \mid M' \in R(M^{12})\}) = \{\mathbf{x} \in \Omega \mid x_a = 0, 0 \leq x_b \leq \kappa_b/\gamma_b\}$. Et puisque $supp(M^{12}) = \{\mathbf{x} \in \Omega \mid x_a = \theta_a^1\}$, nous en déduisons que $\Psi(M^{12})$ est vide. M^{12} est donc un domaine de mode instantané : toutes les solutions arrivant en M^{12} le traversent instantanément.

Nécessité d'un raffinement

En résumé, les résultats présentés ci-dessus montrent (1) qu'il est possible d'étendre la définition des modèles LPM à l'ensemble de l'espace des phases d'une manière systématique et mathématiquement correcte à l'aide d'inclusions différentielles et (2) qu'il est possible de caractériser de façon relativement simple la dynamique locale des systèmes LPM, puisqu'à chaque domaine de mode est associé un ensemble focal vers lequel le système converge de manière monotone, au sens classique pour les domaines réguliers, et avec un sens faible pour les domaines singuliers. Il est alors possible de connaître la dynamique globale des systèmes LPM, souvent très complexe, en juxtaposant les dynamiques locales. Une esquisse du flot dans l'espace des phases de notre exemple est présentée figure 4.5.

C'est sur cette analyse que reposent les travaux précédents [70] et l'implémentation actuelle de l'outil Genetic Network Analyzer (version 5.5) [68]. Toutefois, les prédictions obtenues par cette méthode ne sont pas adaptées aux comparaisons avec les données ex-

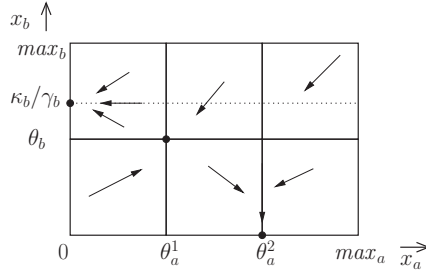


FIG. 4.5 – Portrait de phase global correspondant au système LPM (3.4)-(3.5). Le flot est représenté dans chaque domaine de mode persistant. Les points d'équilibres sont représentés par le symbole \bullet .

périmentales, nécessaires pour tester la validité des modèles. Effectivement, les données expérimentales disponibles donnent des informations sur les sens de variation des concentrations des protéines, c'est-à-dire sur le signe des dérivées des solutions du système LPM. Or, il n'est pas toujours possible de caractériser sans ambiguïté les signes des dérivées des solutions dans les domaines de mode. C'est notamment le cas pour M^{11} , dans la dimension de x_b , pour notre exemple (figure 4.2(b)). Effectivement, le signe de la dérivée dans la dimension de x_b des solutions passant par un point \mathbf{x} de M^{11} dépend de la position de \mathbf{x} par rapport à l'hyperplan $\{\mathbf{x} \in \Omega \mid x_b = \kappa_b/\gamma_b\}$.

Afin d'apporter une solution à ce problème, nous proposons d'utiliser une partition plus fine de l'espace des phases en domaines dans lequel la dynamique du système est qualitativement identique, au sens où les signes des dérivées des solutions sont invariants dans tout le domaine. Ces domaines sont appelés des *domaines de flot*. La définition et la caractérisation de ces domaines de flot fait l'objet des deux sections suivantes.

4.1.3 Partition en domaines de flot

Les domaines de flot, dans lesquels les signes des dérivées des solutions sont invariants, sont obtenus en repartitionnant les domaines de mode selon des hyperplans d'isocline nulle, *i.e.* tels que $0 \in H_i(\mathbf{x})$. Comme précédemment, nous distinguons selon si le domaine de mode est régulier ou singulier.

Partition des domaines de mode réguliers

Considérons un domaine de mode régulier M . M est partitionné en domaines de flot par les hyperplans de dimension $n-1$ définis par $\{\mathbf{x} \in \Omega \mid x_i = \psi_i(M)\}$, $i \in \{1, \dots, n\}$. Plus précisément, M est divisé dans la dimension x_i si et seulement si l'hyperplan $\{\mathbf{x} \in \Omega \mid x_i = \psi_i(M)\}$ intersecte avec M , c'est-à-dire, si et seulement si $\psi_i(M) \in M_i$.

Définition 15 (Domaines de flot : cas régulier). L'ensemble des domaines de flot, \mathcal{D}^M , associé au domaine de mode régulier M est l'ensemble des régions hyperrectangulaires non vides $D = D_1 \times \dots \times D_n \subseteq M$, telles que pour tout $i \in \{1, \dots, n\}$:

- si $\psi_i(M) \notin M_i$, alors $D_i = M_i$,
- sinon, D_i satisfait l'une des équations suivantes :

$$D_i = \{x_i \in M_i \mid x_i < \psi_i(M)\},$$

$$D_i = \{x_i \in M_i \mid x_i = \psi_i(M)\},$$

$$D_i = \{x_i \in M_i \mid \psi_i(M) < x_i\}.$$

La partition des domaines de mode réguliers M^1 et M^{11} est représentée figure 4.6. D'après la définition 15, M^1 n'est divisé ni dans la dimension x_a ($\psi_a(M^1) \notin M_a^1$), ni dans la dimension x_b ($\psi_b(M^1) \notin M_b^1$), comme représenté figure 4.6(a). Ainsi un seul domaine de flot, $D^{1.1}$, correspond au domaine de mode M^1 , avec $D^{1.1} = M^1$ (figure 4.6(c)). La partition de M^{11} , représentée figure 4.6(b), est plus intéressante. Puisque $\psi_b(M^{11}) \in M_b^{11}$, M_b^{11} est divisé en trois ensembles : $\{x_b \in M_b^{11} \mid x_b < \psi_b(M^{11})\}$, $\{x_b \in M_b^{11} \mid x_b = \psi_b(M^{11})\}$ et $\{x_b \in M_b^{11} \mid \psi_b(M^{11}) < x_b\}$. De façon similaire, M_a^{11} est divisé en deux ensembles : $\{x_a \in M_a^{11} \mid x_a = \psi_a(M^{11})\}$ et $\{x_a \in M_a^{11} \mid \psi_a(M^{11}) < x_a\}$. Par conséquent, M^{11} est partitionné en $2 \times 3 = 6$ domaines de flot, $D^{11.1}$, \dots , $D^{11.6}$, représentés figure 4.6(d).

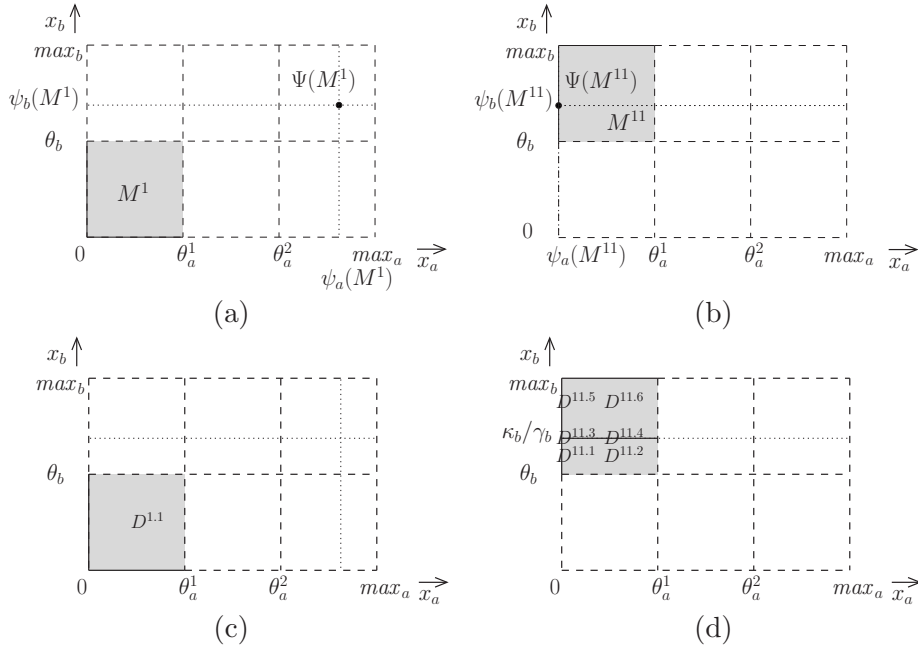


FIG. 4.6 – Partition des domaines de mode réguliers M^1 et M^{11} .

Partition des domaines de mode singuliers

Considérons un domaine de flot singulier M . M est partitionné en domaines de flot par les hyperplans de dimension $n-1$, définis par $\{\mathbf{x} \in \Omega \mid x_i = \psi_i(M')\}$, $M' \in R(M)$, $i \in \{1, \dots, n\}$. Plus précisément, nous définissons l'ensemble des concentrations focales qui partitionnent M dans la dimension x_i , $\Lambda_i(M)$, comme

$$\Lambda_i(M) = \{\psi_i(M') \in M_i \mid M' \in R(M)\}, \quad i \in \{1, \dots, n\}.$$

Nous notons $\lambda_i^1, \dots, \lambda_i^{q_i}$ les éléments de $\Lambda_i(M)$, ordonnés tels que $\lambda_i^1 < \dots < \lambda_i^{q_i}$. Ainsi M est divisé dans la dimension x_i si et seulement si M_i n'est pas un singleton et $\Lambda_i(M) \neq \emptyset$, c'est-à-dire, si et seulement si au moins un des hyperplans $\{\mathbf{x} \in \Omega \mid x_i = \psi_i(M')\}$, $M' \in R(M)$, intersecte avec M_i .

Définition 16 (Domaines de flot : cas singulier). L'ensemble des domaines de flot, \mathcal{D}^M , associé au domaine de mode singulier M est l'ensemble des régions hyperrectangulaires non vides $D = D_1 \times \dots \times D_n \subseteq M$, telles que pour tout $i \in \{1, \dots, n\}$:

- si M_i est un singleton ou $\Lambda_i(M) = \emptyset$, alors $D_i = M_i$,

– sinon, $\Lambda_i(M) = \{\lambda_i^1, \dots, \lambda_i^{q_i}\}$ et D_i satisfait l'une des équations suivantes :

$$D_i = \{x_i \in M_i \mid x_i < \lambda_i^1\},$$

$$D_i = \{x_i \in M_i \mid x_i = \lambda_i^1\},$$

$$D_i = \{x_i \in M_i \mid \lambda_i^1 < x_i < \lambda_i^2\},$$

...

$$D_i = \{x_i \in M_i \mid x_i = \lambda_i^{q_i}\},$$

$$D_i = \{x_i \in M_i \mid \lambda_i^{q_i} < x_i\}.$$

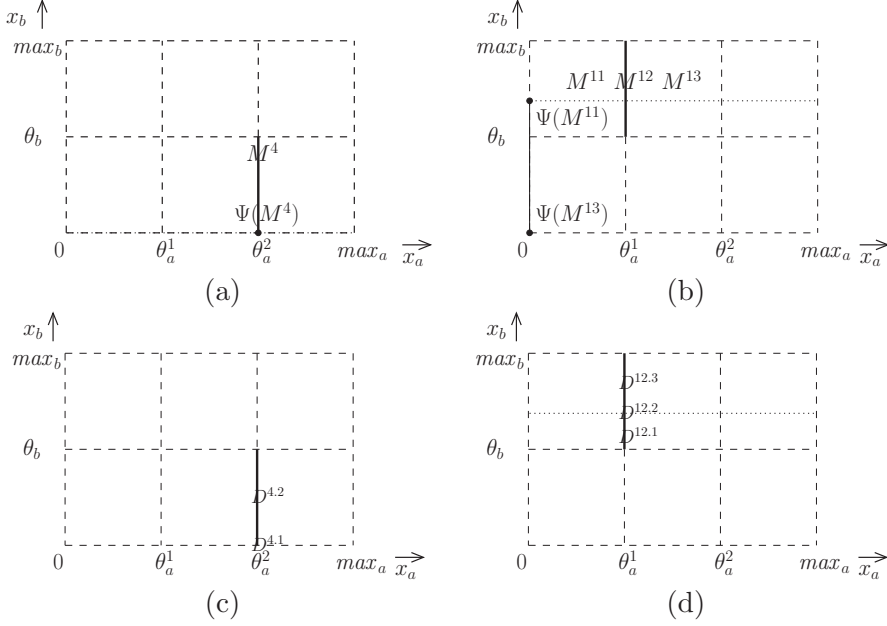


FIG. 4.7 – Partition des domaines de mode singuliers M^4 et M^{12} .

La partition du domaine de mode singulier M^4 est représentée figure 4.7(a). Avec $R(M^4) = \{M^3, M^5\}$ et $\psi_b(M^3) = \psi_b(M^5) = 0 \in M_b^4$, on a $\Lambda_b(M^4) = \{0\}$. Alors, étant donné que M_a^4 est un singleton, on déduit de la définition 16 que M^4 est partitionné en deux domaines de flot : $D^{4.1}$ et $D^{4.2}$ (figure 4.7(c)). La partition du domaine de mode singulier M^{12} est représentée figure 4.7(b). Avec $R(M^{12}) = \{M^{11}, M^{13}\}$, $\psi_b(M^{11}) = \kappa_b/\gamma_b$ et $\psi_b(M^{13}) = 0$, on a $\Lambda_b(M^{12}) = \{\kappa_b/\gamma_b\}$. Alors, étant donné que M_a^{12} est un singleton, on déduit de la définition 16 que M^{12} est partitionné en trois domaines de flot : $D^{12.1}$, $D^{12.2}$ et $D^{12.3}$ (figure 4.7(d)).

L'ensemble des domaines de flot, \mathcal{D} , est l'union de tous les domaines de flot obtenus en partitionnant les domaines de mode :

$$\mathcal{D} = \bigcup_{M \in \mathcal{M}} \mathcal{D}^M.$$

\mathcal{D} est une partition de l'espace des phases Ω . Par définition, chaque domaine de flot est inclus dans un unique domaine de mode. On appelle *mode* la fonction de \mathcal{D} vers \mathcal{M} qui représente cette application surjective : $mode(D) = M$, si et seulement si $D \subseteq M$. Les concepts de domaines instantanés et persistant peuvent être naturellement étendus aux domaines de flot : $\mathcal{D}_{pers} = \{D \mid mode(D) \in \mathcal{M}_{pers}\}$. De façon analogue, on introduit la fonction *flot* de Ω vers \mathcal{D} qui représente l'application surjective qui associe un point de

l'espace des phases au domaine de flot qui le contient : $\text{flot}(\mathbf{x}) = D$, si et seulement si $\mathbf{x} \in D$.

La partition \mathcal{D} de l'espace des phases correspondant au système LPM (3.4)-(3.5) est composée de 27 domaines de flot, $D^{1.1}, \dots, D^{15.1}$. \mathcal{D} est représentée figure 4.8. On a par exemple, $\text{mode}(D^{1.1}) = M^1$ et $\text{mode}(D^{2.1}) = \text{mode}(D^{2.2}) = M^2$. $D^{1.1}, D^{11.1}, \dots, D^{11.6}$ sont des domaines de flot persistants alors que $D^{12.1}, D^{12.2}, D^{12.3}$ sont instantanés.

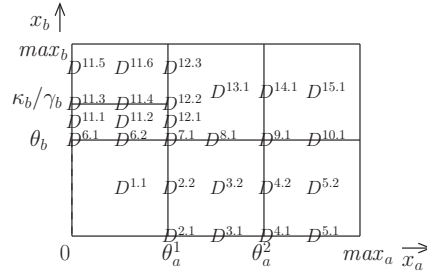


FIG. 4.8 – Partition en domaines de flot de l'espace des phases correspondant au réseau représenté figure 3.1.

4.1.4 Analyse de la dynamique dans les domaines de flot

Dans cette section, nous introduisons la notion de *pattern de signes* des dérivées, caractérisant qualitativement les dérivées des solutions en un point, et nous montrons que ce pattern est identique pour tout point d'un domaine de flot.

Considérons un point \mathbf{x} d'un domaine de flot $D \in \mathcal{D}$ et l'ensemble des solutions $\xi \in \Xi_\Sigma$ dans D passant en \mathbf{x} . On note $S(\mathbf{x}) \in 2^{\{-1,0,1\}^n}$ le pattern de signes des dérivées en \mathbf{x} , c'est-à-dire l'ensemble des vecteurs de signes des dérivées de ces solutions en \mathbf{x} .

Définition 17 (Pattern de signes des dérivées). Le pattern de signes des dérivées en \mathbf{x} , $S(\mathbf{x})$, est défini comme :

$$S(\mathbf{x}) = \{\text{sign}(\dot{\xi}(t_x)) \mid \xi \in \Xi_\Sigma \text{ dans } D, \xi(t_x) = \mathbf{x}, \text{ et } \dot{\xi}(t_x) \in H(\xi(t_x))\}$$

$S(\mathbf{x})$ décrit de façon qualitative la dynamique du système au point \mathbf{x} . Puisque nous utilisons des inclusions différentielles, il peut y avoir des solutions de Σ qui passent en un point \mathbf{x} avec des dérivées différentes, voire avec des signes de dérivées différents. Par conséquent, $S(\mathbf{x})$ peut en toute généralité ne pas être un singleton. Ce cas ne se produit pas pour notre exemple de réseau d'inhibition réciproque.

Notons également que cette définition ne prend pas en compte les solutions de Σ qui traversent D instantanément, ou qui ne satisfont pas (3.8) en \mathbf{x} . En particulier, si \mathbf{x} est dans un domaine de flot instantané D , alors $S(\mathbf{x})$ est vide, puisque par définition des domaines instantanés, toutes les solutions traversent D instantanément. Comme pour $H(\mathbf{x})$, il est possible d'écrire $S(\mathbf{x})$ comme le produit de ses composantes si $S(\mathbf{x})$ n'est pas vide : $S(\mathbf{x}) = S_1(\mathbf{x}) \times \dots \times S_n(\mathbf{x})$. $S_i(\mathbf{x}) \in 2^{\{-1,0,1\}}$ décrit la dynamique au point \mathbf{x} dans la dimension de x_i , $i \in \{1, \dots, n\}$.

Afin d'illustrer cette notion, considérons le domaine de flot $D^{11.4}$ représenté figure 4.8. Toutes les solutions dans $D^{11.4}$ sont strictement décroissantes dans la dimension de x_a et constantes dans la dimension de x_b , puisqu'elles convergent monotoniquement vers l'ensemble focal $\Psi(\text{mode}(D^{11.4})) = \Psi(M^{11})$ (figure 4.6(b)). Donc, pour tout $\mathbf{x} \in D^{11.4}$,

$S(\mathbf{x}) = \{(-1, 0)'\}$. Puisque $D^{12.1}$ est un domaine de flot instantané, toutes les solutions traversent $D^{12.1}$ instantanément, de telle sorte que pour tout $\mathbf{x} \in D^{12.1}$, $S(\mathbf{x}) = \emptyset$.

Le théorème suivant montre que tous les points d'un domaine de flot ont le même pattern de signes de dérivées. Exprimé différemment, le théorème 4 prouve que \mathcal{D} est une partition de Ω qui divise l'espace des phases en régions dans lesquelles la dynamique est qualitativement identique, au sens où le pattern de signes de dérivées est identique partout. Il justifie la terminologie « domaine de flot ».

Théorème 4 (Dynamique qualitativement identique dans les domaines de flot).

Pour tout $D \in \mathcal{D}$ et pour tout $\mathbf{x}, \mathbf{x}' \in D$, $S(\mathbf{x}) = S(\mathbf{x}')$.

Démonstration. Soient D un domaine de flot, et \mathbf{x} et \mathbf{x}' deux points dans D . Si D est un domaine de flot instantané, aucune solution ne reste dans D . Par conséquent, l'ensemble des solutions $\xi \in \Xi_\Sigma$ dans D et passant en \mathbf{x} est vide, et d'après la définition 17, $S(\mathbf{x}) = \emptyset$. Par le même raisonnement, on déduit que $S(\mathbf{x}') = \emptyset$.

Si D n'est pas un domaine de flot instantané, alors D est persistant. Soient $\xi \in \Xi_\Sigma$ et $t_x \geq 0$ tels que ξ est une solution dans D vérifiant $\xi(t_x) = \mathbf{x}$ et $\dot{\xi}(t_x) \in H(\xi(t_x))$. Premièrement, considérons le cas où D_i est un singleton, $i \in \{1, \dots, n\}$. On a alors nécessairement $\dot{\xi}_i(t_x) = 0$, puisque ξ_i est dérivable en t_x et reste constante dans D_i , et donc, $S_i(\mathbf{x}) = \{0\}$. Par le même raisonnement, on déduit que $S_i(\mathbf{x}') = \{0\}$. Supposons maintenant que D_i n'est pas réduit à un singleton, et que $\dot{\xi}_i(t_x) > 0$ (les cas $\dot{\xi}_i(t_x) = 0$ et $\dot{\xi}_i(t_x) < 0$ sont similaires). On montre que $\dot{\xi}(t_x) \in H(\xi(t_x))$ et $\dot{\xi}_i(t_x) > 0$ impliquent l'existence d'un certain $\psi \in \Psi(\text{mode}(D))$ tel que $\psi_i - \xi_i(t_x) > 0$ (voir (4.1) et la définition 13 si $\text{mode}(D)$ est régulier, et (4.2) et la définition 14 si $\text{mode}(D)$ est singulier). Plus précisément, il existe un certain $\psi \in \Psi(\text{mode}(D)) \cap \text{supp}(D)$ satisfaisant les conditions ci-dessus, étant donné que $\Psi(\text{mode}(D))$ est un ensemble hyperrectangulaire et que $\Psi(\text{mode}(D)) \cap \text{supp}(D) \neq \emptyset$ (puisque D est persistant et d'après le lemme 1, annexe A). D'après la définition de la partition de Ω en domaines de flot (définitions 15 et 16), on en déduit qu'il existe un certain $\psi' \in \Psi(\text{mode}(D)) \cap \text{supp}(D)$, éventuellement égal à ψ , tel que $\psi_i - x_i$ et $\psi'_i - x'_i$ ont même signe, c'est-à-dire, tel que $\psi'_i - x'_i > 0$. Alors, en utilisant le lemme 2 (annexe A), on montre qu'il existe $\xi' \in \Xi_\Sigma$ dans $\text{mode}(D)$ tel que $\xi'(t'_x) = \mathbf{x}'$ pour un certain $t'_x \geq 0$, et tel que ξ'_i converge monotoniquement vers ψ'_i . Puisque, $\psi' \in \text{supp}(D)$, ξ' reste dans D en convergeant vers ψ' . On a de plus $\dot{\xi}'_i(t'_x) > 0$. Nous avons donc montré que si $1 \in S_i(\mathbf{x})$, alors $1 \in S_i(\mathbf{x}')$ (définition 17). On démontre de façon similaire que si $s_i \in S_i(\mathbf{x})$, alors $s_i \in S_i(\mathbf{x}')$, pour $s_i \in \{-1, 0\}$. Nous en concluons donc que $S_i(\mathbf{x}) \subseteq S_i(\mathbf{x}')$. Puisque les rôles de \mathbf{x} et \mathbf{x}' sont symétriques, nous concluons que $S_i(\mathbf{x}) = S_i(\mathbf{x}')$. ■

Ce théorème sert de charnière entre l'analyse mathématique des modèles LPM, faite dans cette section, et l'analyse algorithmique, présentée dans les sections suivantes. Effectivement, il justifie la partition proposée dans cette section et sert de base à l'utilisation d'abstraction discrète proposée dans la section suivante.

4.2 Abstraction discrète

Dans cette section, nous utilisons le concept d'*abstraction discrète*, introduite originellement dans la communauté du model checking pour les systèmes discrets, puis généralisée aux systèmes hybrides (voir section 2.1.4). Cette abstraction discrète produit une description qualitative de la dynamique du système dans l'espace des phases. Nous la qualifierons donc également d'*abstraction qualitative*.

Afin d'établir clairement la relation entre le système LPM original et le système abstrait, dit système de transitions LPM qualitatif, nous introduisons dans un premier temps un *système de transitions continu* équivalent, dans un sens défini précisément par la suite, au système LPM original. Puis, le système abstrait est défini comme le *quotient discret* du système de transitions continu, défini à l'aide d'une relation d'équivalence *préservant les propositions*. Cette démarche permet de conserver des propriétés intéressantes du système LPM original (sections 2.1.4 et 2.2.3).

4.2.1 Système de transitions LPM

A tout système LPM Σ , nous associons un système de transitions continu, Σ -TS, défini de telle sorte que Σ et Σ -TS aient des propriétés d'atteignabilités équivalentes. Considérons $\mathbf{x} \in D$ et $\mathbf{x}' \in D'$, avec $D, D' \in \mathcal{D}$, deux domaines de flot. S'il existe une solution ξ de Σ passant en \mathbf{x} à l'instant τ et atteignant \mathbf{x}' à l'instant τ' , sans quitter $D \cup D'$ pendant l'intervalle de temps $[\tau, \tau']$, alors la continuité absolue de ξ implique que D et D' sont soit identiques, soit contigus. Plus précisément, on a l'un des trois cas suivants : $D = D'$, $D \in \partial D'$, ou $D' \in \partial D$. Par conséquent, nous distinguons trois types de transitions continues : *interne*, notées $\mathbf{x} \xrightarrow{int} \mathbf{x}'$, *dimension croissante*, notées $\mathbf{x} \xrightarrow{dim^+} \mathbf{x}'$, et *dimension décroissante*, notées $\mathbf{x} \xrightarrow{dim^-} \mathbf{x}'$. Ces deux derniers termes expriment que la dimension du domaine augmente ou diminue, en allant de D à D' . Cela nous conduit à la définition suivante :

Définition 18 (Système de transitions LPM). Le système de transitions LPM associé au système LPM $\Sigma = (\Omega, \Theta, H)$ est le tuple Σ -TS $= (\Omega, L, \Pi, \rightarrow, \models)$ où :

- Ω est l'espace d'état ou espace des phases.
- $L = \{int, dim^+, dim^-\}$ est un ensemble d'étiquettes représentant les trois types de transitions possibles.
- $\Pi = \{Dsign = S \mid S \in 2^{\{-1,0,1\}^n}\}$ est un ensemble de propositions permettant de décrire qualitativement la dynamique du système.
- \rightarrow est une relation de transition décrivant l'évolution continue du système, définie par $\rightarrow \subseteq \Omega \times L \times \Omega$, telle que $\mathbf{x} \xrightarrow{l} \mathbf{x}'$ si et seulement si il existe $\xi \in \Xi_\Sigma$ et τ, τ' , tels que $0 \leq \tau < \tau'$, $\xi(\tau) = \mathbf{x}$, $\xi(\tau') = \mathbf{x}'$, et
 - si $l = int$, alors $flot(\mathbf{x}) = flot(\mathbf{x}')$ et pour tout $t \in [\tau, \tau']$, on a $\xi(t) \in flot(\mathbf{x})$,
 - si $l = dim^+$, alors $flot(\mathbf{x}) \neq flot(\mathbf{x}')$ et pour tout $t \in (\tau, \tau']$, on a $\xi(t) \in flot(\mathbf{x}')$,
 - si $l = dim^-$, alors $flot(\mathbf{x}) \neq flot(\mathbf{x}')$ et pour tout $t \in [\tau, \tau')$, on a $\xi(t) \in flot(\mathbf{x})$.
- \models est la relation de satisfaction des propositions de Π , définie par $\models \subseteq \Omega \times \Pi$, telle que $\mathbf{x} \models Dsign = S$ si et seulement si $S = S(\mathbf{x})$.

La relation de satisfaction \models associe à chaque point \mathbf{x} de l'espace des phases la description qualitative de la dynamique en \mathbf{x} introduite à la section précédente.

Une séquence de points de Ω , $(\mathbf{x}^0, \dots, \mathbf{x}^m)$, $m \geq 0$, est définie comme une *exécution* de Σ -TS si et seulement si une séquence de transitions permet d'atteindre \mathbf{x}^m à partir de \mathbf{x}^0 , c'est-à-dire, si et seulement si pour tout $i \in \{0, \dots, m-1\}$, il existe $l \in L$ tel que $\mathbf{x}^i \xrightarrow{l} \mathbf{x}^{i+1}$.

Le théorème suivant montre qu'un système LPM Σ et son système de transitions LPM associé Σ -TS ont des propriétés d'atteignabilité équivalentes. La démonstration de la proposition est donnée dans l'annexe A.

Proposition 2 (Atteignabilité équivalente). Pour tout $\mathbf{x}, \mathbf{x}' \in \Omega$, il existe une solution ξ de Σ et τ, τ' tels que $0 \leq \tau \leq \tau'$, $\xi(\tau) = \mathbf{x}$, et $\xi(\tau') = \mathbf{x}'$ si et seulement si il existe une exécution $(\mathbf{x}^0, \dots, \mathbf{x}^m)$ de Σ -TS telle que $\mathbf{x}^0 = \mathbf{x}$ et $\mathbf{x}^m = \mathbf{x}'$.

Afin d'illustrer ces notions, considérons la solution ξ du système LPM Σ décrit par (3.4)-(3.5) représentée figure 4.9(a). ξ passe par les points \mathbf{x}^0 , \mathbf{x}^1 , \mathbf{x}^2 , et \mathbf{x}^3 aux instants respectifs 0, τ^1 , τ^2 , et τ^3 . Soit Σ -TS le système de transitions LPM associé à Σ . D'après la définition 18, et puisque \mathbf{x}^0 et \mathbf{x}^1 sont dans le même domaine de flot, $D^{1.1}$, il existe une transition de type *int* de \mathbf{x}^0 vers \mathbf{x}^1 . Également, d'après la définition 18, et puisque $\text{flot}(\mathbf{x}^1) = D^{1.1}$, $\text{flot}(\mathbf{x}^2) = D^{2.2}$ et $\xi(t) \in D^{1.1}$, pour tout $t \in [\tau^1, \tau^2)$, il existe une transition de type *dim*⁻ de \mathbf{x}^1 vers \mathbf{x}^2 . De même, il existe une transition *dim*⁻ de \mathbf{x}^0 vers \mathbf{x}^2 , et une transition *dim*⁺ de \mathbf{x}^2 vers \mathbf{x}^3 . La séquence de points $(\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3)$ est une exécution de Σ -TS. Enfin, d'après la convergence monotone vers $\Psi(\text{mode}(D^{1.1}))$ des solutions de Σ dans $D^{1.1}$, on peut déduire que $S(\mathbf{x}^1) = \{(1, 1)'\}$, et donc, en utilisant la définition 18, que $\mathbf{x}^1 \models \text{Dsign} = \{(1, 1)'\}$ (voir figure 4.9(b)).

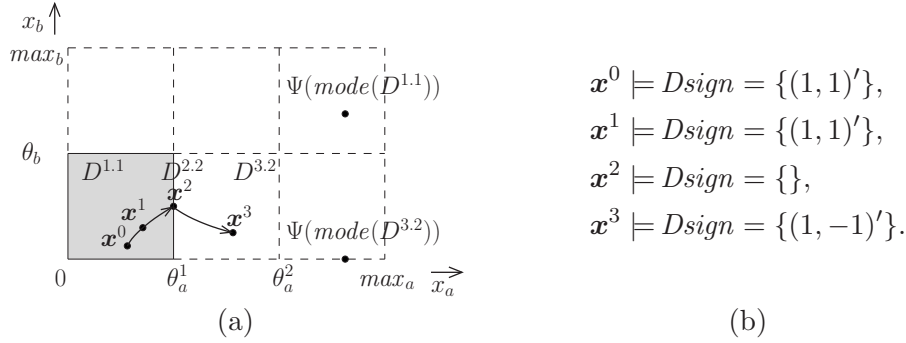


FIG. 4.9 – (a) Exemple d'une trajectoire partant de $D^{1.1}$ et atteignant $D^{3.2}$ en passant par les points \mathbf{x}^0 , \mathbf{x}^1 , \mathbf{x}^2 et \mathbf{x}^3 aux instants respectifs 0, τ^1 , τ^2 et τ^3 . (b) Relation de satisfaction pour les points \mathbf{x}^0 , \mathbf{x}^1 , \mathbf{x}^2 et \mathbf{x}^3 .

4.2.2 Système de transitions LPM qualitatif

Le système de transitions LPM qualitatif est défini à l'aide de la relation d'équivalence, $\sim_\Omega \subseteq \Omega \times \Omega$, induite par la partition \mathcal{D} de l'espace des phases : $\mathbf{x} \sim_\Omega \mathbf{x}'$ si et seulement si $\text{flot}(\mathbf{x}) = \text{flot}(\mathbf{x}')$.

On vérifie simplement que \sim_Ω préserve les propositions (section 2.1). Effectivement, soient \mathbf{x} et \mathbf{x}' deux points équivalents, c'est-à-dire deux points d'un même domaine de flot, alors d'après le théorème 4, on a $S(\mathbf{x}) = S(\mathbf{x}')$.

Le système de transitions LPM qualitatif, Σ -QTS, associé au système de transitions LPM Σ -TS, est alors défini comme le système de transitions quotient de Σ -TS, étant donné la relation d'équivalence préservant les propositions \sim_Ω (section 2.1). Σ -QTS est une abstraction discrète de Σ -TS.

Définition 19 (Système de transitions LPM qualitatif). Le système de transitions LPM qualitatif associé au système de transitions LPM Σ -TS = $(\Omega, L, \Pi, \rightarrow, \models)$ est le tuple Σ -QTS = $(\Omega/\sim_\Omega, L, \Pi, \rightarrow_{\sim_\Omega}, \models_{\sim_\Omega})$.

La proposition 3 découle directement des définitions précédentes.

Proposition 3 (Système de transitions LPM qualitatif).

Soit $\Sigma\text{-QTS} = (\Omega/\sim_\Omega, L, \Pi, \rightarrow_{\sim_\Omega}, \models_{\sim_\Omega})$ le système de transitions LPM qualitatif associé au système de transitions LPM $\Sigma\text{-TS} = (\Omega, L, \Pi, \rightarrow, \models)$. Alors,

- $\Omega/\sim_\Omega = \mathcal{D}$.
- $\rightarrow_{\sim_\Omega} \subseteq \mathcal{D} \times L \times \mathcal{D}$, telle que $D \xrightarrow{l}_{\sim_\Omega} D'$ si et seulement si il existe $\xi \in \Xi_\Sigma$ et τ, τ' tels que $0 \leq \tau < \tau'$, $\xi(\tau) \in D$, $\xi(\tau') \in D'$, et
 - si $l = \text{int}$, alors $D = D'$ et pour tout $t \in [\tau, \tau']$, on a $\xi(t) \in D$,
 - si $l = \text{dim}^+$, alors $D \neq D'$ et pour tout $t \in (\tau, \tau']$, on a $\xi(t) \in D'$,
 - si $l = \text{dim}^-$, alors $D \neq D'$ et pour tout $t \in [\tau, \tau')$, on a $\xi(t) \in D$.
- $\models_{\sim_\Omega} \subseteq \mathcal{D} \times \Pi$, telle que $D \models_{\sim_\Omega} \text{Dsign} = S$ si et seulement si pour tout $\mathbf{x} \in D$, $S(\mathbf{x}) = S$.

Démonstration. $\Sigma\text{-QTS}$ est défini comme le système de transitions quotient de $\Sigma\text{-TS}$ par la relation d'équivalence \sim_Ω . D'après la définition 3, $\Sigma\text{-QTS}$ est alors donné par $(\Omega/\sim_\Omega, L, \Pi, \rightarrow_{\sim_\Omega}, \models_{\sim_\Omega})$, où

- Ω/\sim_Ω est l'espace d'état quotient,
- $\rightarrow_{\sim_\Omega} : \Omega/\sim_\Omega \times L \times \Omega/\sim_\Omega$ tel que $P \xrightarrow{l}_{\sim_\Omega} P'$ si et seulement si il existe $\mathbf{x} \in P$, et $\mathbf{x}' \in P'$ tels que $\mathbf{x} \xrightarrow{l} \mathbf{x}'$, et
- $\models_{\sim_\Omega} : \Omega/\sim_\Omega \times \Pi$ tel que $P \models_{\sim_\Omega} \pi$ si et seulement si il existe $\mathbf{x} \in P$ tel que $\mathbf{x} \models \pi$.

Premièrement, Ω/\sim_Ω est égal à \mathcal{D} puisque \sim_Ω est définie comme la relation d'équivalence sur Ω induite par \mathcal{D} , donc la première partie de la proposition 3 est vraie. La seconde partie de la proposition est vraie par définition de $\mathbf{x} \xrightarrow{l}_{\sim_\Omega} \mathbf{x}'$ dans $\Sigma\text{-TS}$ (définition 18). Enfin, par définition, $P \models_{\sim_\Omega} \pi$ si et seulement si il existe $\mathbf{x} \in P$ tel que $\mathbf{x} \models \pi$. Plus précisément, étant donné que π est de la forme $\text{Dsign} = S$, $P \models_{\sim_\Omega} \pi$ si et seulement si il existe $\mathbf{x} \in P$ tel que $S(\mathbf{x}) = S$ (définition 18). On obtient alors immédiatement la dernière partie de la proposition 3 à l'aide du théorème 4. \blacksquare

Contrairement à $\Sigma\text{-TS}$, $\Sigma\text{-QTS}$ est un système de transitions *fini*, car son espace d'état \mathcal{D} est un ensemble fini. Il s'agit d'une propriété essentielle, parce qu'elle permet de calculer $\Sigma\text{-QTS}$, de le représenter et de l'analyser à l'aide de techniques de model checking classiques.

Les transitions étiquetées par dim^+ ou dim^- connectent deux domaines de flot différents ($D \neq D'$, d'après la proposition 3). Elles correspondent donc à une évolution continue durant laquelle un changement de domaine de flot a lieu. A l'inverse, une transition *int* est une boucle, c'est-à-dire une transition d'un état sur lui-même, et correspond à une évolution continue durant laquelle le système reste dans un même domaine de flot. Comme pour $\Sigma\text{-TS}$, nous définissons toute séquence de domaines de flot (D^0, \dots, D^m) , $m \geq 0$ comme une exécution de $\Sigma\text{-QTS}$ si et seulement si pour tout $i \in \{0, \dots, m-1\}$, il existe $l \in L$ tel que $D^i \xrightarrow{l}_{\sim_\Omega} D^{i+1}$.

La relation de satisfaction \models_{\sim_Ω} associe à chaque domaine de flot $D \in \mathcal{D}$, une description qualitative de la dynamique dans ce domaine. C'est pourquoi, les états de $\Sigma\text{-QTS}$ sont également appelés des *états qualitatifs*, et $\Sigma\text{-QTS}$, une *abstraction qualitative* de $\Sigma\text{-TS}$. De plus, la relation \models_{\sim_Ω} associe à chaque exécution une description qualitative de l'évolution du système au cours du temps. Par conséquent, les exécutions de $\Sigma\text{-QTS}$ sont également appelées des *comportements qualitatifs* [67, 69, 70]. $\Sigma\text{-QTS}$ peut être représenté par un graphe dirigé, $G = (\mathcal{D}, \rightarrow_{\sim_\Omega})$, appelé *graphe de transitions d'état*. Les chemins du graphe sont des exécutions.

Considérons à nouveau la situation représenté figure 4.9. L'application de la proposition 3 donne que $D^{1.1}$ peut être décrit qualitativement par $D^{1.1} \models_{\sim_{\Omega}} Dsign = \{1, 1\}$. Egalement, d'après la proposition 3, il existe une transition int sur $D^{1.1}$, une transition dim^- de $D^{1.1}$ vers $D^{2.2}$ et une transition dim^+ de $D^{2.2}$ vers $D^{3.2}$. Le graphe de transitions d'état $G = (\mathcal{D}, \rightarrow_{\sim_{\Omega}})$ correspondant au système LPM (3.4)-(3.5) est représenté figure 4.10(a). Afin de faciliter la lisibilité, les étiquettes des transitions entre domaines de flots sont omises. $(D^{1.1}, D^{2.2}, D^{3.2}, D^{4.2}, D^{4.1})$ et $(D^{1.1}, D^{6.2}, D^{11.2}, D^{11.3})$ sont deux exécutions, décrites dans la figure 4.10(b).

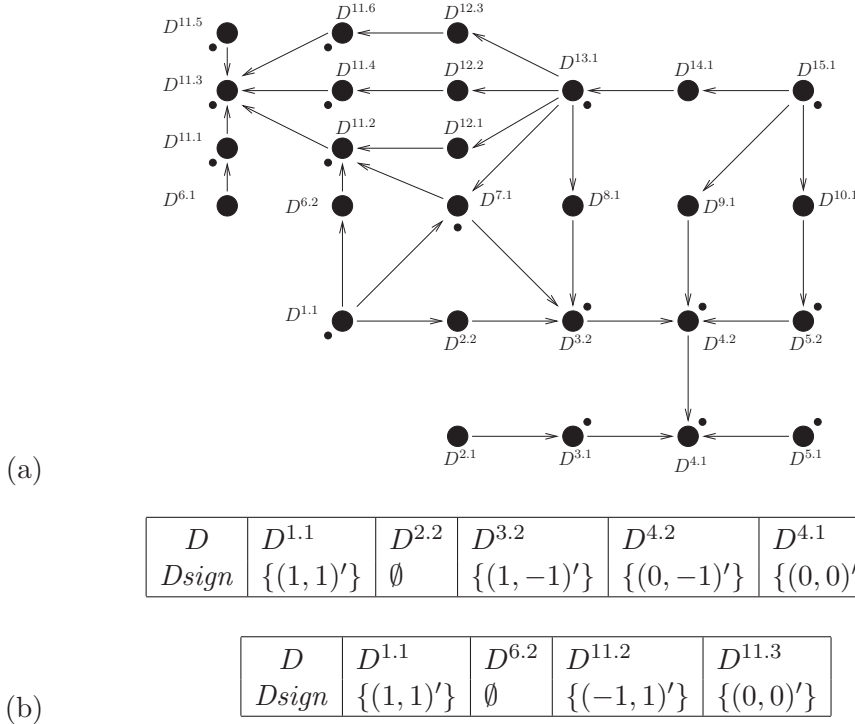


FIG. 4.10 – (a) Graphe de transitions d'états $G = (\mathcal{D}, \rightarrow_{\sim_{\Omega}})$ associé au système LPM (3.4)-(3.5). Les symboles \bullet représentent des boucles sur les états. (b) Description de deux exécutions $(D^{1.1}, D^{2.2}, D^{3.2}, D^{4.2}, D^{4.1})$ et $(D^{1.1}, D^{6.2}, D^{11.2}, D^{11.3})$.

Par définition, un système de transitions quotient *simule* le système original (section 2.1.4). Donc Σ -QTS simule Σ -TS. La réciproque étant en général fautive, Σ -QTS et Σ -TS ne sont en général pas bisimilaires. Ceci implique en particulier que si il existe une exécution $(\mathbf{x}^0, \dots, \mathbf{x}^m)$ de Σ -TS, alors il existe une exécution (D^0, \dots, D^m) de Σ -QTS telle que $\mathbf{x}^i \in D^i$, pour tout $i \in \{0, \dots, m\}$. La simulation proposée abstrait notamment les aspects quantitatifs du temps. Dans la terminologie de [71, 220], la simulation est dite « time-abstracting », au sens où entre deux transitions, on sait seulement qu'un certain temps passe, mais pas exactement combien.

Enfin, nous pouvons formaliser à l'aide des propositions 1 et 2 (sections 2.1.4 et 4.2.1) la relation existant entre Σ et Σ -QTS. Considérons une solution ξ du système LPM Σ définie sur un intervalle de temps (fini ou infini) I et passant par la séquence ordonnée dans le temps de domaines de flot (D^0, \dots, D^m) durant l'intervalle de temps I . Alors, (D^0, \dots, D^m) est une exécution de Σ -QTS. A l'inverse, il est possible qu'aucune solution ne passe par une séquence de domaines de flot correspondante à une exécution de Σ -QTS.

On montre ainsi que Σ -QTS est une *approximation conservative* de Σ . Ceci justifie formellement l'utilisation de Σ -QTS pour étudier la dynamique qualitative de Σ .

Théorème 5 (Approximation conservative). Σ -QTS est une approximation conservative de Σ .

Démonstration. Pour tout $\mathbf{x}, \mathbf{x}' \in \Omega$, nous voulons montrer que si il existe une solution ξ de Σ et τ, τ' tels que $0 \leq \tau \leq \tau'$, $\xi(\tau) = \mathbf{x}$, et $\xi(\tau') = \mathbf{x}'$, alors il existe une exécution (D^0, \dots, D^m) de Σ -QTS telle que $\mathbf{x} \in D^0$ et $\mathbf{x}' \in D^m$. Considérons deux points $\mathbf{x}, \mathbf{x}' \in \Omega$, une solution $\xi \in \Xi_\Sigma$ et τ, τ' tels que $0 \leq \tau \leq \tau'$, $\xi(\tau) = \mathbf{x}$, et $\xi(\tau') = \mathbf{x}'$. Alors, d'après la proposition 2, il existe une exécution $(\mathbf{x}^0, \dots, \mathbf{x}^m)$ de Σ -TS telle que $\mathbf{x}^0 = \mathbf{x}$ et $\mathbf{x}^m = \mathbf{x}'$. Puisque Σ -QTS est un système de transitions quotient de Σ -TS par une relation d'équivalence, Σ -QTS simule Σ -TS (proposition 1). Par une conséquence immédiate de la définition de simulation (section 2.1.4), on obtient alors l'existence d'une exécution (D^0, \dots, D^m) de Σ -QTS telle que pour tout $i \in \{1, \dots, m\}$, $\mathbf{x}^i \in D^i$, et donc telle que $\mathbf{x} \in D^0$ et $\mathbf{x}' \in D^m$. ■

Considérons à nouveau la solution ξ dont la trajectoire durant l'intervalle de temps $I = [0, \tau^3]$ est représentée figure 4.9(a). Elle passe par $D^{1.1}$, $D^{2.2}$, et $D^{3.2}$, et on peut vérifier que $(D^{1.1}, D^{2.2}, D^{3.2})$ est effectivement une exécution de Σ -QTS, c'est-à-dire un chemin de G (figure 4.10). A l'inverse, il est possible qu'une exécution de Σ -QTS ne corresponde à aucune exécution de Σ -TS, c'est-à-dire à aucune solution de Σ .

4.3 Modèles qualitatifs

La dynamique d'un système LPM Σ dépend naturellement des valeurs de ses paramètres, données par les vecteurs κ , γ et θ introduits à la section 3.1.2. Néanmoins, deux systèmes LPM qui diffèrent par leurs valeurs de paramètre peuvent correspondre à un même système de transitions LPM qualitatif, et par conséquent, satisfaire les mêmes propriétés qualitatives. Nous introduisons une seconde relation d'équivalence, $\sim_\Gamma \subseteq \Gamma \times \Gamma$, définie sur l'espace des paramètres Γ du système LPM Σ telle que deux vecteurs de paramètre \mathbf{p} et \mathbf{p}' sont équivalents si les systèmes de transitions LPM qualitatifs correspondants sont identiques, ou plus précisément, isomorphes. Deux systèmes de transitions LPM qualitatifs sont isomorphes s'ils ont des domaines de flot ayant même connectivité et satisfaisant les mêmes propositions. Formellement, l'isomorphie de systèmes de transitions LPM qualitatifs est définie par l'existence d'une bijection entre leurs domaines de flot préservant les relations $\rightarrow_{\sim_\Omega}$ et \models_{\sim_Ω} . L'espace des paramètres quotient, Γ/\sim_Γ , représente alors une partition de l'espace des paramètres en régions dans lesquelles tous les systèmes LPM correspondent à un même système de transitions LPM qualitatif.

Dans la suite, nous proposons une classe de contraintes d'inégalité sur les paramètres définissant des régions $P \subseteq \Gamma$, telle que pour tout $\mathbf{p}, \mathbf{p}' \in P$, on a $\mathbf{p} \sim_\Gamma \mathbf{p}'$. Soient Θ_i et Λ_i les ensembles de concentrations de seuil et de concentrations focales pour le gène i : $\Theta_i = \{\theta_i^1, \dots, \theta_i^{p_i}\}$ et $\Lambda_i = \{\psi_i(M) \mid M \in \mathcal{M}_r\}$. Rappelons que les concentrations focales $\psi_i(M)$ dépendent directement des paramètres κ et γ , puisqu'elles sont des ratios de (sommés de) constantes de taux de synthèse et de (sommés de) constantes de taux de dégradation. On définit alors des *contraintes d'inégalité sur les paramètres* comme suit.

Définition 20 (Contraintes d'inégalité sur les paramètres). Pour tout $i \in \{1, \dots, n\}$, les contraintes d'inégalité sur les paramètres de la variable x_i consistent en un ordre total sur $\Theta_i \cup \Lambda_i$.

A tout système LPM, nous associons un ensemble de contraintes d'inégalité sur les paramètres, donné par des contraintes d'inégalité sur les paramètres pour chacune de ses variables. Un ensemble de contraintes d'inégalité sur les paramètres définit une région P dans l'espace des paramètres. Le théorème suivant montre qu'une telle région est incluse dans une seule classe d'équivalence de Γ/\sim_Γ , c'est-à-dire, qu'un ensemble de contraintes d'inégalité sur les paramètres définit une classe de modèles LPM correspondant à un même système de transitions LPM qualitatif, et satisfaisant donc tous les mêmes propriétés qualitatives.

Théorème 6 (Equivalence de modèles LPM). Soit P une région de l'espace des paramètres Γ définie par un ensemble de contraintes d'inégalité sur les paramètres d'un système LPM Σ . Alors, il existe $Q \in \Gamma/\sim_\Gamma$ tel que $P \subseteq Q$.

Démonstration.² Nous voulons montrer que deux systèmes LPM qui ne diffèrent que par la valeur de leurs paramètres, \mathbf{p} et \mathbf{p}' , sont associés à des systèmes de transitions LPM qualitatifs isomorphes si \mathbf{p} et \mathbf{p}' satisfont le même ensemble de contraintes d'inégalité sur les paramètres (définition 20). On peut facilement vérifier que, si \mathbf{p} et \mathbf{p}' satisfont le même ensemble de contraintes d'inégalité sur les paramètres, \mathbf{p} et \mathbf{p}' satisfont exactement les mêmes conditions pour le calcul de la partition \mathcal{D} (définitions 12, 15, et 16, section 4.1), de la relation \models_{\sim_Ω} (propositions 4 et 5, section 4.4.2), et de la relation $\rightarrow_{\sim_\Omega}$ (propositions 6, 7, et 8, section 4.4.3), lors du calcul des systèmes de transitions LPM qualitatifs associés aux deux systèmes LPM. Par conséquent, les systèmes de transitions LPM qualitatifs associés aux deux systèmes LPM sont isomorphes. ■

Un *modèle qualitatif* est alors défini comme un système LPM associé avec un ensemble de contraintes d'inégalité sur ses paramètres.

Pour le système LPM (3.4)-(3.5), on a $\Theta_a \cup \Lambda_a = \{\theta_a^1, \theta_a^2, 0, \kappa_a/\gamma_a\}$ et $\Theta_b \cup \Lambda_b = \{\theta_b, 0, \kappa_b/\gamma_b\}$. Les valeurs numériques données à la section 3.1.2 pour ce système LPM, $\kappa = (18, 7)$, $\gamma = (2, 1)$ et $\theta = (4, 8, 4)$, correspondent aux contraintes d'inégalité suivantes :

$$0 < \theta_a^1 < \theta_a^2 < \kappa_a/\gamma_a < \max_a, \quad (4.3)$$

$$0 < \theta_b < \kappa_b/\gamma_b < \max_b. \quad (4.4)$$

Par conséquent, le système de transitions LPM qualitatif représenté figure 4.10(a) décrit non seulement la dynamique du modèle LPM donné à la section 3.1.2, mais décrit également la dynamique de tous les modèles LPM définis par les équations (3.4)-(3.5) et dont les paramètres satisfont les contraintes d'inégalité (4.3)-(4.4).

Deux remarques essentielles doivent être faites ici. Premièrement, ces contraintes d'inégalité sur les paramètres ont un sens biologique et peuvent en général, et ce contrairement aux valeurs numériques des paramètres, être spécifiées à partir des données disponibles dans la littérature expérimentale. Deuxièmement, il est possible, à l'aide de cette seule information qualitative, de calculer le système de transitions LPM qualitatif associé au modèle qualitatif. Ce second point est développé dans la section suivante.

²Une preuve similaire est donnée dans [70].

4.4 Calcul symbolique du système de transitions LPM qualitatif

Dans cette section, nous considérons un modèle qualitatif et notre objectif est de calculer le système de transitions LPM qualitatif associé. Nous proposons des règles simples pour réaliser ce calcul de façon symbolique et efficace, même dans un espace d'état de dimension élevée. Deux caractéristiques essentielles des systèmes étudiés rendent un tel calcul réalisable. Tout d'abord, tous les ensembles à calculer – domaines de mode, de flot, hyperplans support, ensembles focaux, \dots – sont hyperrectangulaires. Par conséquent, ils peuvent être calculés séparément dans chaque dimension. Également, toutes les bornes de ces ensembles correspondent soit à des concentrations de seuil, soit à des concentrations focales, c'est-à-dire à des éléments de $\Theta_i \cup \Lambda_i$, $i \in \{1, \dots, n\}$, dont l'ordre est donné par les contraintes d'inégalité sur les paramètres de la variable x_i . Ainsi, les calculs se réduisent en pratique à de simples comparaisons entre éléments de $\Theta_i \cup \Lambda_i$, $i \in \{1, \dots, n\}$.

Dans les trois sections suivantes, nous montrons successivement comment calculer la partition \mathcal{D} de l'espace des phases, la caractérisation qualitative de la dynamique dans les domaines donnée par la relation \models_{\sim_Ω} , et les transitions entre domaines, $\rightarrow_{\sim_\Omega}$. L et Π sont respectivement égaux aux constantes $\{int, dim^+, dim^-\}$ et $\{Dsign = S \mid S \in 2^{\{-1,0,1\}^n}\}$.

4.4.1 Calcul de l'ensemble des domaines

Nous montrons ici que l'on peut successivement calculer la partition de l'espace des phases en domaines de mode, \mathcal{M} , puis en domaines de flot, \mathcal{D} .

Les contraintes d'inégalité sur les paramètres donnent directement l'ordre des concentrations de seuil, $\Theta_i = \{\theta_i^1, \dots, \theta_i^{p_i}\}$, $i \in \{1, \dots, n\}$. On peut alors déterminer la partition \mathcal{M} à l'aide de la définition 12. \mathcal{D} est alors obtenue en repartitionnant les domaines de mode.

Si $M \in \mathcal{M}$ est un domaine de mode régulier, l'évaluation de \mathbf{f} et de \mathbf{g} dans M donne $\boldsymbol{\mu}^M$ et $\boldsymbol{\nu}^M$, et par conséquent $\boldsymbol{\psi}(M)$, où $\psi_i(M) = \mu_i^M / \nu_i^M$, $i \in \{1, \dots, n\}$ (définition 13). Alors, dans chaque dimension i , les contraintes d'inégalité sur les paramètres permettent de déterminer si $\psi_i(M) \in M_i$, puisque $\psi_i(M)$ est une concentration focale et M_i est borné par deux concentrations de seuil. La partition de M peut alors être calculée en appliquant la définition 15.

Dans l'autre cas, $M \in \mathcal{M}$ est un domaine de mode singulier. Connaissant la partition \mathcal{M} , on peut déterminer $R(M)$, l'ensemble des domaines de mode réguliers qui ont M dans leur frontière. Puis pour chaque domaine de mode régulier M' de $R(M)$, on évalue comme précédemment $\boldsymbol{\psi}(M')$, et pour chaque dimension i , si $\psi_i(M') \in M_i$. $\Lambda_i(M)$ est donc connu et la partition de M peut alors être calculée en appliquant la définition 16. Nous obtenons alors \mathcal{D} , l'ensemble de tous les domaines de flot.

4.4.2 Calcul des propriétés des domaines

Calculer la relation de satisfaction \models_{\sim_Ω} revient, d'après la proposition 3, à calculer $S(\mathbf{x})$, $\mathbf{x} \in D$, pour tout $D \in \mathcal{D}$. Nous proposons dans un premier temps des règles permettant de caractériser $S(\mathbf{x})$, puis nous montrerons comment ces règles peuvent être évaluées symboliquement.

Considérons un point \mathbf{x} dans un domaine de flot persistant $D \in \mathcal{D}_{pers}$ et une solution $\boldsymbol{\xi}$ dans D passant en \mathbf{x} à l'instant t_x , satisfaisant $\dot{\boldsymbol{\xi}}(t_x) \in H(\boldsymbol{\xi}(t_x))$. Considérons également

chaque dimension i séparément, $i \in \{1, \dots, n\}$. Nous limitons ici notre analyse au cas où $x_i \notin \Psi_i(\text{mode}(D))$, pour lequel la convergence de ξ_i vers $\Psi_i(\text{mode}(D))$ est monotone (voir section 4.1.2). Dans le cas contraire, l'analyse est plus complexe, mais la proposition suivante reste vraie (voir démonstration). Si D_i est un singleton, alors ξ glisse le long du support de D , et par conséquent, $\dot{\xi}_i(t_x) = 0$. Donc $S_i(\mathbf{x}) = \{0\}$ (définition 17). Si D_i n'est pas un singleton, alors la monotonie de la convergence de ξ_i vers $\Psi_i(\text{mode}(D))$ implique que la dérivée de ξ_i à l'instant t_x est de même signe que la différence $\psi_i - x_i$, avec $\psi \in \Psi(\text{mode}(D))$. Ces raisonnements intuitifs sont formalisés dans la proposition suivante.

Proposition 4 (Calcul de $D\text{sign}$: domaine persistant). Soit $D \in \mathcal{D}$ un domaine de flot persistant et $S \in 2^{\{-1,0,1\}^n}$ tel que $D \models_{\sim_\Omega} D\text{sign} = S$. Alors, $S = S_1 \times \dots \times S_n$ et pour tout $i \in \{1, \dots, n\}$,

- si D_i est un singleton, alors $S_i = \{0\}$,
- si D_i n'est pas un singleton, alors
 - $-1 \in S_i$ si et seulement si il existe $\mathbf{x} \in D$ et $\psi \in \Psi(\text{mode}(D))$ tels que $\psi_i - x_i < 0$,
 - $0 \in S_i$ si et seulement si il existe $\mathbf{x} \in D$ et $\psi \in \Psi(\text{mode}(D))$ tels que $\psi_i - x_i = 0$,
 - $1 \in S_i$ si et seulement si il existe $\mathbf{x} \in D$ et $\psi \in \Psi(\text{mode}(D))$ tels que $\psi_i - x_i > 0$.

Démonstration. Soit $D \in \mathcal{D}$ un domaine de flot persistant. Rappelons que, par définition du système de transitions LPM qualitatif (définition 19), $D \models_{\sim_\Omega} D\text{sign} = S$ est équivalent à $S = S(\mathbf{x})$, pour un certain $\mathbf{x} \in D$. Considérons $i \in \{1, \dots, n\}$ tel que D_i est un singleton. Alors, comme montré dans la preuve du théorème 4, on a, pour tout $\mathbf{x} \in D$, $S_i(\mathbf{x}) = \{0\}$. Maintenant, considérons $i \in \{1, \dots, n\}$ tel que D_i ne soit pas réduit à un singleton. Nous ne considérerons que le cas où $-1 \in S_i$, les cas $1 \in S_i$ et $0 \in S_i$ étant démontrés de façon similaire.

(Réciproque) Considérons $\mathbf{x} \in D$ et $\psi \in \Psi(\text{mode}(D))$, et supposons que $\psi_i - x_i < 0$. Plus précisément, nous supposons que $\psi \in \Psi(\text{mode}(D)) \cap \text{supp}(D)$, l'existence d'un tel ψ découlant de la persistance de D et du lemme 1 (annexe A). On a alors d'après le lemme 2 (annexe A), l'existence d'une solution $\xi \in \Xi_\Sigma$ dans $\text{mode}(D)$ passant par \mathbf{x} à l'instant t_x telle que ξ_i converge monotoniquement vers ψ_i , et par conséquent, telle que $\dot{\xi}_i(t_x) < 0$. Puisque $\psi \in \text{supp}(D)$, ξ reste dans D en convergeant vers ψ . Nous concluons donc que $-1 \in S_i(\mathbf{x})$ (définition 17).

(Sens direct) Par contraposition, nous supposons que pour tout $\mathbf{x} \in D$ et pour tout $\psi \in \Psi(\text{mode}(D))$, on a $\psi_i - x_i \geq 0$. Etant donné que D_i n'est pas réduit à un singleton, et la définition de la partition de Ω en domaines de flot (définitions 15 et 16), l'inégalité précédente est stricte ($\psi_i - x_i > 0$). Nous pouvons alors appliquer le lemme 1 qui implique, d'après la monotonie de la convergence des solutions dans D vers $\Psi(\text{mode}(D))$, que toutes les solutions dans D sont croissantes dans la dimension de x_i . Donc, $-1 \notin S_i(\mathbf{x})$ pour tout $\mathbf{x} \in D$, ce qui termine notre raisonnement par contraposition. \blacksquare

Si D est un domaine de flot instantané, toutes les solutions traversent D instantanément, de telle sorte que d'après la définition 17, $S(\mathbf{x}) = \emptyset$ pour $\mathbf{x} \in D$.

Proposition 5 (Calcul de $D\text{sign}$: domaine instantané). Soit $D \in \mathcal{D}$ un domaine de flot instantané et $S \in 2^{\{-1,0,1\}^n}$ tel que $D \models_{\sim_\Omega} D\text{sign} = S$. Alors, $S = \emptyset$.

La démonstration de la proposition est donnée dans l'annexe A.

Il nous reste à montrer comment ces règles peuvent être évaluées symboliquement, à l'aide des contraintes d'inégalité sur les paramètres.

Afin de pouvoir appliquer les propositions précédentes, il faut premièrement calculer $\Psi(\text{mode}(D))$. Nous avons vu dans la section 4.4.1 comment calculer l'ensemble focal des domaines de mode réguliers. Nous ne considérons donc que le cas où $M = \text{mode}(D)$ est singulier. Afin de calculer $\Psi(M)$, on calcule, pour chaque dimension i , l'intersection de $\text{supp}_i(M)$ avec $P_i(M) = \overline{\text{rect}_i}(\{\psi(M') \mid M' \in R(M)\}) = [\min_{M' \in R(M)} \psi_i(M'), \max_{M' \in R(M)} \psi_i(M')]$ (définition 14). Puisque l'on connaît l'ordre des éléments dans Λ_i , on peut déterminer les bornes de $P_i(M)$. Si M_i n'est pas un singleton, alors $\text{supp}_i(M) = \Omega_i$, et l'intersection entre $\text{supp}_i(M)$ et $P_i(M)$ est égale à ce dernier. Sinon, $\text{supp}_i(M) = \{\theta_i\}$, pour un certain $\theta_i \in \Theta_i$, et l'intersection entre $\text{supp}_i(M)$ et $P_i(M)$ est soit vide soit égale à $\text{supp}_i(M)$, selon si $\theta_i \in P_i(M)$, ce qui peut être déterminé à partir de l'ordre des éléments de $\Theta_i \cup \Lambda_i$. On peut donc calculer symboliquement l'ensemble focal de tous les domaines de mode (réguliers et singuliers).

Comment alors calculer la relation \models_{\sim_Ω} en connaissant les ensembles focaux ? Considérons un domaine de flot $D \in \mathcal{M}$. Si $\Psi(\text{mode}(D))$ est vide, D est domaine de flot instantané et on a $D \models_{\sim_\Omega} \text{Design} = \emptyset$ (proposition 5). Sinon, on applique la proposition 4. On a $D \models_{\sim_\Omega} \text{Design} = S$, avec $S = S_1 \times \dots \times S_n$, et S_i à déterminer pour tout $i \in \{1, \dots, n\}$. Si D_i est un singleton, alors $S_i = \{0\}$. Sinon, $\Psi_i(M') = [\min_{M' \in R(M)} \psi_i(M'), \max_{M' \in R(M)} \psi_i(M')]$ et on compare les bornes de $\Psi_i(M')$ avec celles de D_i , afin de déterminer si $-1, 0$, et 1 appartiennent à S_i .

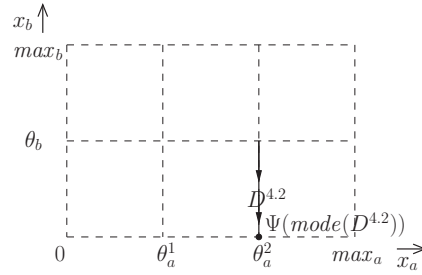


FIG. 4.11 – Calcul du pattern de signes de dérivées dans $D^{4.2}$: $D^{4.2} \models_{\sim_\Omega} \text{Design} = \{(0, -1)\}$.

L'utilisation de la proposition 4 est illustrée figure 4.11. Considérons le domaine de flot $D^{4.2}$ de notre exemple. $\Psi(\text{mode}(D^{4.2}))$ est égal à l'intersection de $\text{supp}(\text{mode}(D^{4.2})) = \{\mathbf{x} \in \Omega \mid x_a = \theta_a^2\}$ et de $\overline{\text{rect}}(\{\psi(M^3), \psi(M^5)\}) = \{\mathbf{x} \in \Omega \mid 0 \leq x_a \leq \kappa_a/\gamma_a \text{ et } x_b = 0\}$ (voir section 4.1.2). Puisque d'après (4.3)-(4.4), $0 < \theta_a^2 < \kappa_a/\gamma_a$, on a $\Psi(\text{mode}(D^{4.2})) = \{(\theta_a^2, 0)'\}$. $\Psi(\text{mode}(D^{4.2}))$ n'est pas vide, donc $D^{4.2}$ est persistant et on peut appliquer la proposition 4. On a alors $D^{4.2} \models_{\sim_\Omega} \text{Design} = S$, avec $S = S_a \times S_b$, et S_a et S_b déterminés comme suit. $D_a^{4.2}$ est un singleton, donc $S_a = \{0\}$. Par ailleurs, $D_b^{4.2}$ n'est pas un singleton, et on a $\psi_b - x_b < 0$, pour tout $\mathbf{x} \in D^{4.2}$ et $\psi \in \Psi(\text{mode}(D^{4.2}))$, de telle sorte que $S_b = \{-1\}$. On peut donc conclure que $D^{4.2} \models_{\sim_\Omega} \text{Design} = \{(0, -1)\}$.

4.4.3 Calcul des transitions entre domaines

Nous avons vu à la section 4.2.2 que l'existence d'une transition entre deux domaines de flot D et D' correspond à l'existence d'une solution ξ qui passe en un point $\mathbf{x} \in D$ à un instant τ puis en un point $\mathbf{x}' \in D'$ à un instant τ' , sans quitter $D \cup D'$ durant l'intervalle de temps $[\tau, \tau']$. Nous sommes alors dans l'un des trois cas suivants : $D = D'$, $D' \subseteq \partial D$, ou $D \subseteq \partial D'$, correspondant respectivement à l'existence d'une transition de type *int*,

dim^- , ou dim^+ . Dans cette section, nous proposons successivement pour chaque type de transition des règles pour leur calcul et nous montrons comment ces règles peuvent être évaluées symboliquement.

Les transitions de type *int* sont facilement caractérisées. D'après la proposition 3, l'existence d'une transition de type *int* sur un domaine D correspond à l'existence d'au moins une solution restant dans D plus longtemps qu'un seul instant. Ainsi tous les domaines de flot persistants, et seulement ceux-ci, ont une boucle *int*, c'est-à-dire, une transition sur eux-même. La démonstration de la proposition est donnée dans l'annexe A.

Proposition 6 (Calcul des transitions *int*). Soient D et D' deux domaines de flot. $D \xrightarrow{int} \sim_{\Omega} D'$ si et seulement si

1. $D = D'$, et
2. $D \in \mathcal{D}_{pers}$

Nous avons vu à la section précédente que l'on peut, pour tout domaine de flot D , déterminer symboliquement l'ensemble focal $\Psi(mode(D))$. On peut alors appliquer directement la proposition ci-dessus, en utilisant $D \in \mathcal{D}_{pers}$ si et seulement si $\Psi(mode(D))$ est vide.

Considérons maintenant les transitions de type dim^+ . D'après la proposition 3, il existe une transition de type dim^+ de D vers D' si et seulement si il existe une solution ξ passant en $x \in D$ à un instant τ puis en $x' \in D'$ à un instant τ' , et restant dans D' pendant l'intervalle de temps $(\tau, \tau']$. Comme vu précédemment, cela implique que $D \subseteq \partial D'$. Considérons alors chaque dimension i séparément, $i \in \{1, \dots, n\}$. Nous limitons ici notre analyse au cas où $x_i \notin \Psi_i(mode(D'))$, pour lequel on sait que la convergence de ξ_i vers $\Psi_i(mode(D'))$ est monotone pendant l'intervalle de temps $(\tau, \tau']$ (voir section 4.1.2). Dans le cas contraire, l'analyse est plus complexe, mais la proposition suivante reste vraie (voir la démonstration). D'après la monotonie de ξ_i , la dérivée de ξ_i sur l'intervalle (τ, τ') est de même signe que la différence $x'_i - x_i$. Également, puisque ξ_i converge vers $\Psi_i(mode(D'))$ pendant l'intervalle (τ, τ') , la dérivée de ξ_i sur l'intervalle (τ, τ') est aussi du signe de $\psi'_i - x'_i$, pour $\psi' \in \Psi(mode(D'))$. On a donc $(\psi'_i - x'_i)(x'_i - x_i) > 0$. On peut montrer que cette condition n'est nécessaire que dans les dimensions i telles que D_i , mais pas D'_i , est un singleton. Ces raisonnements intuitifs sont formalisés dans la proposition suivante.

Proposition 7 (Calcul des transitions dim^+). Soient D et D' deux domaines de flot. $D \xrightarrow{dim^+} \sim_{\Omega} D'$ si et seulement si

1. $D \subseteq \partial D'$, et
2. il existe $x \in D$, $x' \in D'$, et $\psi' \in \Psi(mode(D'))$, tels que pour tout $i \in \{1, \dots, n\}$ pour lesquels D_i , mais pas D'_i , est un singleton, on a

$$(\psi'_i - x'_i)(x'_i - x_i) > 0.$$

Démonstration.

(Sens direct) D'après la proposition 3, une transition de type dim^+ de D vers D' implique l'existence d'une solution ξ absolument continue sur l'intervalle de temps $[\tau, \tau']$ satisfaisant $\xi(\tau) \in D$ et $\xi(t) \in D'$ pour tout $t \in (\tau, \tau']$. Il existe donc au moins une dimension $i \in \{1, \dots, n\}$ telle que D_i est un intervalle fermé, et D'_i un intervalle ouvert. Par conséquent, $D \subseteq \partial D'$, et la condition 1 doit être satisfaite.

Supposons que la condition 2 ne soit pas vérifiée. Alors, il existe une dimension $i \in \{1, \dots, n\}$ telle que D_i , mais pas D'_i , est un singleton et telle que $(\psi'_i - x'_i)(x'_i - x_i) \leq 0$ pour tout $\mathbf{x} \in D$, $\mathbf{x}' \in D'$, et $\boldsymbol{\psi}' \in \Psi(\text{mode}(D'))$. Si $x'_i < x_i$, alors on a $\psi'_i \geq x'_i$. Étant donné que D'_i n'est pas un singleton, et la définition de la partition de Ω en domaines de flot (définitions 15 et 16), l'inégalité est stricte. De façon similaire, si $x'_i > x_i$, alors on a $\psi'_i < x'_i$. Dans les deux cas, $D'_i \cap \Psi_i(\text{mode}(D')) = \emptyset$, et nous pouvons appliquer le lemme 1 (annexe A), qui implique que pour toutes les solutions $\boldsymbol{\xi} \in \Xi_\Sigma$ dans D' , on a ξ_i qui converge monotoniquement vers $\Psi_i(\text{mode}(D'))$. Par conséquent, étant donné les positions relatives de x_i , x'_i et ψ'_i , aucune solution ne peut entrer dans D' à partir de D , et donc il ne peut y avoir de transition de D vers D' . La condition 2 doit donc être satisfaite.

(Réciproque) Nous définissons pour tout i , ξ_i , partant de D_i et entrant dans D'_i , et nous prouvons que $\boldsymbol{\xi} = (\xi_1, \dots, \xi_n)'$ est une solution qui satisfait les conditions de la proposition 3 pour l'existence d'une transition de type dim^+ .

Si D'_i est un singleton, alors $D_i = D'_i = \{x_i \in \Omega_i \mid x_i = \lambda_i\}$, pour un certain $\lambda_i \in \Lambda_i \cup \Theta_i$, puisque $D \subseteq \partial D'$. Alors, nous définissons $\xi_i(t) = \lambda_i$, pour tout $t \geq 0$. Si ni D'_i ni D_i ne sont des singletons, alors d'après la définition de la partition, $D_i \subseteq D'_i$ et, puisque D'_i est un intervalle ouvert, quelque soit la solution $\boldsymbol{\xi}$ de Σ dans D' vérifiant $\xi_i(0) \in D_i$, ξ_i ne traverse pas D'_i instantanément. Le dernier cas possible correspond à D_i , mais pas D'_i , est un singleton. Supposons à nouveau que $D_i = \{x_i \in \Omega_i \mid x_i = \lambda_i\}$. Si $x'_i < \lambda_i$, alors $\psi'_i < x'_i$ doit être vrai pour un certain $\boldsymbol{\psi}' \in \Psi(\text{mode}(D'))$. Cela implique que pour un certain $M \in \mathcal{M}_r$, $\psi_i(M) < \lambda_i$, avec $M = \text{mode}(D')$, si $\text{mode}(D')$ est un domaine de mode régulier, ou $M \in R(\text{mode}(D'))$, sinon. De façon similaire, si $x'_i > \lambda_i$, alors il existe $M \in \mathcal{M}_r$, tel que $\psi_i(M) > \lambda_i$. Nous définissons alors ξ_i comme $\xi_i(t) = \psi_i(M) + (\lambda_i - \psi_i(M))e^{-\nu_i^M t}$. Soit alors $\boldsymbol{\xi}$ définie par $\boldsymbol{\xi} = (\xi_1, \dots, \xi_n)'$ sur $[0, \tau]$, avec τ est choisi suffisamment petit pour que $\boldsymbol{\xi}$ reste dans D' pendant l'intervalle de temps $(0, \tau]$. On vérifie alors facilement que $\boldsymbol{\xi}$ est une solution de Σ et qu'elle satisfait les conditions de la proposition 3 pour l'existence d'une transition de type dim^+ . ■

Il nous faut maintenant montrer comment, étant donnés deux domaines de flot D et D' tels que $D \subseteq \partial D'$, on peut évaluer symboliquement la proposition 7 à partir des contraintes d'inégalité sur les paramètres. Considérons les dimensions i telles que D_i , mais pas D'_i , est un singleton. Alors, il existe $\lambda_i, \lambda'_i, \lambda''_i \in \Theta_i \cup \Lambda_i$, tels que $D_i = \{\lambda'_i\}$ et soit $D'_i = (\lambda_i, \lambda'_i)$ soit $D'_i = (\lambda'_i, \lambda''_i)$, selon si D' est situé dans la dimension x_i , au-dessous ou au-dessus de D .³ Supposons que D'_i s'écrive $(\lambda'_i, \lambda''_i)$, l'autre cas étant symétrique. Alors, pour tout $\mathbf{x} \in D$ et $\mathbf{x}' \in D'$, $(x'_i - x_i) > 0$. Donc pour tester l'existence d'une transition, il faut tester l'existence d'un $\boldsymbol{\psi}' \in \Psi(\text{mode}(D'))$ tel que $(\psi'_i - x'_i)$ soit positif pour un certain $\mathbf{x}' \in D'$. Ceci est en fait équivalent à l'existence de M' tel que $\psi_i(M') > \lambda'_i$, avec $M' = \text{mode}(D)$ ou $M' \in R(\text{mode}(D'))$, selon si $\text{mode}(D)$ est régulier ou pas, ce qui peut être déterminé à partir de l'ordre des éléments de $\Theta_i \cup \Lambda_i$.

Considérons la situation représentée figure 4.12(a) pour notre exemple de réseau d'inhibition réciproque. Soit $\boldsymbol{\xi}$ une solution satisfaisant $\boldsymbol{\xi}(\tau) \in D^{2.2}$ et $\boldsymbol{\xi}(t) \in D^{3.2}$, pour $t \in (\tau, \tau']$ et $\tau, \tau' \geq 0$. On vérifie que $D^{2.2} \subseteq \partial D^{3.2}$. Également, la seule dimension i pour laquelle $D_i^{2.2}$, mais pas $D_i^{3.2}$, est un singleton, est la dimension de x_a . On a pour tout $\mathbf{x} \in D^{2.2}$ et $\mathbf{x}' \in D^{3.2}$, $x'_a - x_a > 0$. De plus, étant donné que, d'après (4.3)-(4.4), $\theta_a^2 < \kappa_a/\gamma_a$, on a pour tout $\mathbf{x}' \in D^{3.2}$ et $\boldsymbol{\psi}' \in \Psi(\text{mode}(D^{3.2}))$, $\psi'_a - x'_a > 0$. On peut donc conclure qu'il existe une transition dim^+ de $D^{2.2}$ vers $D^{3.2}$. À l'aide de la proposition 7, on

³ D'_i peut inclure la borne λ_i ou λ''_i , dans certains cas particuliers, sans que cela ne change notre analyse.

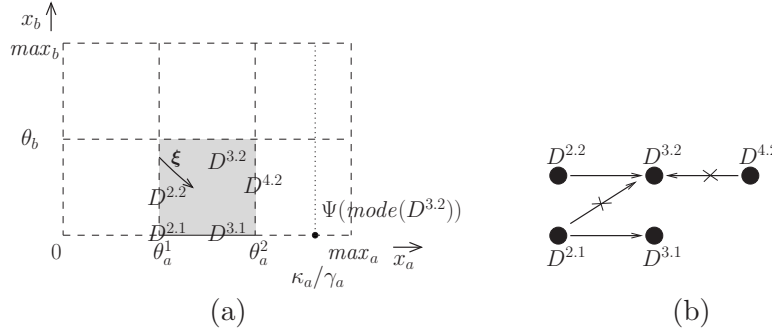


FIG. 4.12 – Calcul des transitions de type \dim^+ entre domaines de flot. (a) Représentation d'une trajectoire passant de $D^{2.2}$ à $D^{3.2}$. (b) Représentation de quelques transitions de type \dim^+ , possibles ou impossibles. Les transitions impossibles sont marquées d'une croix.

peut également exclure l'existence d'une transition de $D^{4.2}$ vers $D^{3.2}$, puisque pour tout $\mathbf{x} \in D^{4.2}$, $\mathbf{x}' \in D^{3.2}$ et $\boldsymbol{\psi}' \in \Psi(\text{mode}(D^{3.2}))$, on a $x'_a - x_a < 0$ tandis que $\psi'_a - x'_a > 0$. De façon similaire, on peut conclure à l'existence d'une transition \dim^+ de $D^{2.1}$ vers $D^{3.1}$, mais pas vers $D^{3.2}$, ce qui est cohérent avec le flot dans $D^{3.2}$. Ces transitions, possibles ou impossibles, sont représentées figure 4.12(b).

Considérons enfin le dernier type de transition : \dim^- . D'après la proposition 3, il existe une transition de type \dim^- de D vers D' si et seulement si il existe une solution $\boldsymbol{\xi}$ passant en $\mathbf{x} \in D$ à un instant $\tau \in \mathbb{R}_{\geq 0}$ puis en $\mathbf{x}' \in D'$ à un instant $\tau' \in \mathbb{R}_{> 0} \cup \{\infty\}$, et restant dans D pendant l'intervalle de temps $[\tau, \tau')$. Comme vu précédemment, cela implique que $D' \subseteq \partial D$. Distinguons les transitions qui ont lieu en temps fini ($\tau' \in \mathbb{R}_{> 0}$) de celles qui n'ont lieu qu'après un temps infini ($\tau' = \{\infty\}$). De façon tout à fait analogue à la proposition 7, on peut montrer qu'il existe une transition en temps fini de D vers D' si et seulement si $D' \subseteq \partial D$ et il existe $\mathbf{x} \in D$, $\mathbf{x}' \in D'$, et $\boldsymbol{\psi} \in \Psi(\text{mode}(D))$, tels que $(\psi_i - x'_i)(x'_i - x_i) > 0$, pour tout $i \in \{1, \dots, n\}$ pour lesquels D'_i , mais pas D_i , est un singleton. A l'inverse, si $\tau' = \infty$, alors $\boldsymbol{\xi}$, restant dans D pendant l'intervalle de temps $[\tau, \infty)$, converge vers $\Psi(\text{mode}(D))$, et par conséquent, atteint asymptotiquement à la fois D' (définition de la transition) et $\Psi(\text{mode}(D))$ (convergence), ce qui implique que $D' \cap \Psi(\text{mode}(D))$ n'est pas vide. Ces raisonnements intuitifs sont formalisés dans la proposition suivante, dont la preuve est donnée dans l'annexe A.

Proposition 8 (Calcul des transitions \dim^-). Soient D et D' deux domaines de flot. $D \xrightarrow{\dim^-} \sim_{\Omega} D'$ si et seulement si

1. $D' \subseteq \partial D$, et
2. il existe $\mathbf{x} \in D$, $\mathbf{x}' \in D'$, et $\boldsymbol{\psi} \in \Psi(\text{mode}(D))$, tels que
 - (a) pour tout $i \in \{1, \dots, n\}$ pour lesquels D'_i , mais pas D_i , est un singleton, on a

$$(\psi_i - x'_i)(x'_i - x_i) > 0,$$

ou (b) on a

$$\mathbf{x}' = \boldsymbol{\psi}.$$

Comment, étant donnés deux domaines de flot D et D' tels que $D' \subseteq \partial D$, peut-on évaluer symboliquement la proposition 8? L'évaluation de la condition 2(a) peut être

réalisée de manière analogue à celle de la condition 2 de la proposition 7, présentée plus haut. Afin d'évaluer la condition 2(b), on teste si l'intersection de D' et de $\Psi(\text{mode}(D))$ est vide ou, de manière équivalente, s'il existe $i \in \{1, \dots, n\}$ tel que $D'_i \cap \Psi_i(\text{mode}(D)) = \emptyset$, ce qui peut être déterminé à partir de l'ordre des éléments de $\Theta_i \cup \Lambda_i$.

Considérons à nouveau l'exemple du réseau d'inhibition réciproque représenté figure 3.1. La figure 4.13 représente deux trajectoires du système dans l'espace des phases de notre exemple. Soit ξ^1 une solution satisfaisant $\xi^1(t) \in D^{1.1}$, pour $t \in [\tau, \tau')$ et $\xi^1(\tau') \in D^{2.2}$. Etant donné que $D^{2.2} \in \partial D^{1.1}$, on peut tester l'existence d'une transition dim^- de $D^{1.1}$ vers $D^{2.2}$. Etant donné que la seule dimension i pour laquelle $D_i^{2.2}$, mais pas $D_i^{1.1}$ est un singleton est la dimension de x_a , et que dans cette dimension, $\psi_a - x'_a > 0$ et $x'_a - x_a > 0$, pour tout $x \in D^{1.1}$, $x' \in D^{2.2}$, et $\psi \in \Psi(\text{mode}(D^{1.1}))$, on conclut qu'il y a effectivement une transition dim^- de $D^{1.1}$ vers $D^{2.2}$. On peut montrer de manière similaire, qu'à l'inverse, il n'existe pas de transition dim^- de $D^{1.1}$ vers $D^{2.2}$. Une telle transition serait en contradiction avec la direction du flot dans $D^{1.1}$.

Soit ξ^{11} une solution satisfaisant $\xi^{11}(t) \in D^{11.6}$, pour $t \in [\tau, \infty)$ et $\lim_{t \rightarrow \infty} \xi^{11}(t) \in D^{11.3}$. On peut vérifier que $D^{11.3} \in \partial D^{11.6}$. On peut vérifier également que la condition 2(a) de la proposition 8 n'est pas satisfaite, puisque pour tout $x' \in D^{11.3}$ et $\psi \in \Psi(\text{mode}(D^{11.6}))$, $\psi_i - x'_i = 0$, $i \in \{a, b\}$ (figure 4.13(a)). Par contre, puisque $D^{11.3}$ et $\Psi(\text{mode}(D^{11.6}))$ intersectent (et sont même égaux), la condition 2(b) est satisfaite. Il y a par conséquent une transition dim^- de $D^{11.6}$ vers $D^{11.3}$. On peut vérifier que si l'on considère une transition de $D^{11.6}$ vers $D^{11.5}$, ni la condition 2(a), ni la condition 2(b), ne sont satisfaites. Il n'y a donc pas de transition. Une telle transition serait en contradiction avec la convergence asymptotique des solutions dans $D^{11.6}$ vers $\Psi(\text{mode}(D^{11.6})) = D^{11.3}$.

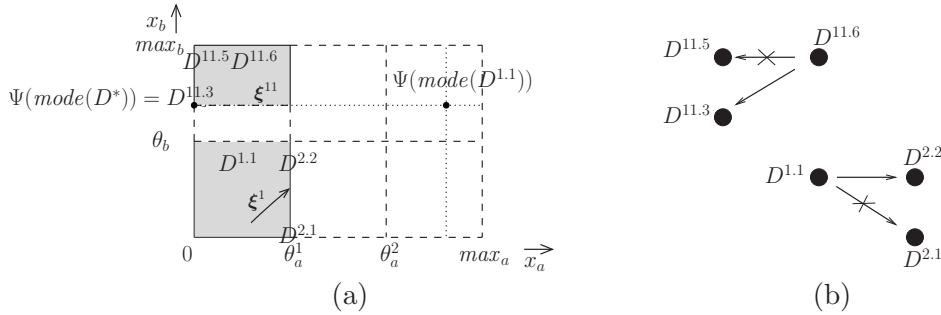


FIG. 4.13 – Calcul des transitions de type dim^- entre domaines de flot. (a) Représentation de deux trajectoires, ξ^1 passant en temps fini de $D^{1.1}$ à $D^{2.2}$, et ξ^{11} passant en temps infini de $D^{11.6}$ à $D^{11.3}$. D^* représente $D^{11.3}$, $D^{11.5}$ et $D^{11.6}$. (b) Représentation de quelques transitions de type dim^- , possibles ou impossibles. Les transitions impossibles sont marquées d'une croix.

En résumé, la méthode présentée dans ce chapitre pour l'analyse qualitative de modèles LPM repose sur trois points importants. Premièrement, nous définissons un système de transitions purement discret, appelé système de transitions LPM qualitatif, qui décrit qualitativement la dynamique du système dans l'espace des phases et qui est une approximation conservatrice du système LPM original. Deuxièmement, nous introduisons la notion de modèle qualitatif constitué d'un système LPM et d'un ensemble de contraintes d'égalité sur les paramètres. Un tel modèle décrit une classe de modèles LPM, qui diffèrent

par leurs valeurs de paramètres, mais qui correspondent à un même système de transitions LPM qualitatif, et qui par conséquent, satisfont les mêmes propriétés qualitatives. De plus, ces contraintes d'inégalité peuvent en général être spécifiées à partir des données disponibles dans la littérature. Enfin, nous proposons des règles simples pour le calcul symbolique du système de transitions LPM qualitatif associé à un modèle qualitatif. Ces règles exploitent les propriétés favorables des systèmes LPM que l'on considère.

Comme nous allons le voir dans la partie Applications, l'implémentation de ces règles dans un outil informatique permet d'analyser des modèles qualitatifs comprenant un nombre de variables élevé. Les systèmes de transitions LPM qualitatifs ainsi obtenus peuvent contenir un grand nombre d'états et de transitions entre états. Dans ce cas, l'analyse manuelle d'un tel système de transitions est difficile et sujette à l'erreur. Afin d'apporter une solution à ce problème, nous proposons de combiner cette méthode avec des techniques de model checking, qui nous permettront d'analyser les propriétés dynamiques de grands modèles de façon systématique et efficace.

5 Vérification par model checking

Nous avons proposé dans le chapitre précédent une méthode pour l'analyse symbolique de modèles qualitatifs de réseaux de régulation génique. Cette méthode génère un système de transitions discret, appelé système de transitions LPM qualitatif, qui décrit qualitativement la dynamique du système dans l'espace des phases et qui est une approximation conservative du système LPM original. Les réseaux de régulation génique biologiquement intéressants sont souvent de grande taille, et leur analyse par la méthode proposée au chapitre précédent génère des systèmes de transitions LPM qualitatifs pouvant contenir un grand nombre d'états et de transitions entre états. L'analyse manuelle de ces systèmes de transitions est difficile et sujette à l'erreur. Dans cette section, nous montrons que l'utilisation de techniques de model checking permet de proposer une solution à ce problème.

Dans le cadre formel que nous proposons, le système est représenté sous la forme d'une structure de Kripke, et les propriétés sont exprimées dans une logique temporelle (voir section 2.2). Aussi, nous présentons successivement comment reformuler le système de transitions LPM qualitatif en une structure de Kripke, puis comment exprimer des propriétés biologiquement pertinentes sur la dynamique du système dans des logiques temporelles, et enfin, comment à l'aide de model checkers, il est possible de tester si les prédictions du modèles vérifient les propriétés. Nous utiliserons deux logiques temporelles, complémentaires sur les aspects de la facilité d'utilisation, de l'efficacité de la vérification, et de l'expressivité : CTL et le μ -calcul (section 2.2).

Les résultats présentés dans ce chapitre ont fait l'objet des publications [21, 23].

5.1 Formulation du système de transitions LPM qualitatif en structure de Kripke

Le système de transitions LPM qualitatif est reformulé en une structure de Kripke afin de permettre son analyse par model checking. L'ensemble des propositions atomiques servant à décrire l'état du système doit être défini préalablement. Avec la méthode présentée dans le chapitre précédent, un état du système correspond à une région dans l'espace des phases et est caractérisé par les signes des dérivées des solutions dans cette région. Ainsi, les propositions atomiques que nous considérons portent soit sur les valeurs des concentrations des protéines du réseau, soit sur leurs signes de dérivée.

Définition 21 (Propositions atomiques). L'ensemble des propositions atomiques associé au système de transitions LPM qualitatif Σ -QTS est :

$$AP = \{x_i \sim \lambda_i \mid \sim \in \{>, \geq, =, \leq, <\}, \lambda_i \in \Theta_i \cup \Lambda_i \text{ et } i \in \{1, \dots, n\}\} \cup \{Dsign_i = \sigma_i \mid \sigma_i \in 2^{\{-1,0,1\}} \text{ et } i \in \{1, \dots, n\}\}. \quad (5.1)$$

La structure de Kripke associée au système de transitions LPM qualitatif Σ -QTS est alors définie comme suit.

Définition 22 (Structure de Kripke). Soit Σ -QTS = $(\Omega/\sim_\Omega, L, \Pi, \rightarrow_{\sim_\Omega}, \models_{\sim_\Omega})$ un système de transitions LPM qualitatif. Alors la structure de Kripke associée à Σ -QTS est Σ -KS = $(\mathcal{D}, \rightarrow_{KS}, \models_{KS})$, telle que

- $\rightarrow_{KS} \subseteq \mathcal{D} \times \mathcal{D}$ est une relation de transition telle que $D \rightarrow_{KS} D'$ si et seulement si il existe $l \in L$ tel que $D \xrightarrow{l}_{\sim_\Omega} D'$,
- $\models_{KS} \subseteq \mathcal{D} \times AP$ est une relation de satisfaction telle que $D \models_{KS} p$ si et seulement si
 - si p est de type ' $x_i \sim \lambda_i$ ', alors pour tout $\mathbf{x} \in D$, on a $x_i \sim \lambda_i$,
 - si p est de type ' $Dsign_i = \sigma_i$ ', alors on a $\sigma_i = S_i$, où $S_i \in 2^{\{-1,0,1\}}$ tel que $S = S_1 \times \dots \times S_n$ et $D \models_{\sim_\Omega} Dsign = S$.

Informellement, Σ -KS correspond simplement à Σ -QTS dans lequel on a enlevé les étiquettes des transitions, int , dim^+ et dim^- , et on a complété la relation de satisfaction avec des propriétés du type ' $x_i \sim \lambda_i$ ', avec $\sim \in \{>, \geq, =, \leq, <\}$, dont l'interprétation est usuelle. On peut vérifier que l'existence pour tout $\mathbf{x}_0 \in \Omega$ d'au moins une solution $\xi \in \Xi_\Sigma$ sur un certain intervalle de temps $[0, \tau]$, $\tau > 0$ satisfaisant $\xi(0) = \mathbf{x}_0$ (section 3.2) garantit que la relation de transition \rightarrow_{KS} est bien totale.

L'ensemble des propositions atomiques correspondant au système de transitions LPM qualitatif de l'exemple du réseau d'inhibition réciproque présenté dans la section 3.1 est

$$AP = \{x_a > 0, x_a > \theta_a^1, \dots, x_b < max_b, Dsign_a = \{\}, \dots, Dsign_b = \{-1, 0, 1\}\}.$$

La structure de Kripke associée au système de transitions LPM qualitatif représenté figure 4.10(a) et correspondant à notre exemple, est représentée figure 5.1.

$$\begin{aligned} \Sigma\text{-KS} &= (\mathcal{D}, \rightarrow_{KS}, \models_{KS}), \text{ avec} \\ - \mathcal{D} &= \{D^{1.1}, D^{2.1}, D^{2.2}, \dots, D^{15.1}\}, \\ - \rightarrow_{KS} &= \{(D^{1.1}, D^{1.1}), (D^{1.1}, D^{2.2}), (D^{1.1}, D^{6.2}), (D^{1.1}, D^{7.1}), (D^{2.1}, D^{3.1}), (D^{2.2}, D^{3.2}), \\ &\quad \dots, (D^{15.1}, D^{14.1})\}, \\ - \models_{KS} &= \{(D^{1.1}, x_a \geq 0), (D^{1.1}, x_a < \theta_a^1), \dots, (D^{1.1}, Dsign_a = \{1\}), (D^{1.1}, Dsign_b = \{1\}), \\ &\quad \dots, (D^{2.1}, x_a = \theta_a^1), (D^{2.1}, x_b = 0), \dots, (D^{2.1}, Dsign_a = \{\}), (D^{2.1}, Dsign_b = \{\}), \\ &\quad \dots, (D^{15.1}, x_a > \theta_a^2), \dots, (D^{15.1}, Dsign_a = \{-1\}), (D^{15.1}, Dsign_b = \{-1\})\}. \end{aligned}$$

FIG. 5.1 – Structure de Kripke Σ -KS associée au système de transitions LPM qualitatif Σ -QTS représenté figure 4.10.

5.2 Formulation de propriétés dynamiques en logique temporelle

L'objectif de cette section est de montrer la diversité des questions biologiquement pertinentes sur la dynamique du système que l'on peut formuler, à l'aide de logiques temporelles. Dans la suite, nous donnons quelques exemples de propriétés, en précisant leur signification biologique et leur formalisation en logique temporelle, en μ -calcul ou en CTL. On pourra se reporter à [46] pour un analyse similaire, et à [75] pour une perspective plus générale. L'impossibilité de pouvoir exprimer certaines propriétés importantes en CTL illustre bien la nécessité de pouvoir utiliser des logiques très expressives, comme le μ -calcul.

5.2.1 Propriétés exprimables en CTL et en μ -calcul

Dans cette section, nous présentons d'abord des propriétés portant sur l'*état* du système, exprimées simplement à l'aide des opérateurs classiques de logique propositionnelle, puis nous présentons des propriétés portant sur le *comportement* du système, pour lesquelles l'utilisation des opérateurs de logique temporelle est nécessaire.

Propriétés sur les états

L'existence de solutions dont la dérivée dans une dimension i , est positive, négative ou nulle est caractérisée par les propositions suivantes :

$$\begin{aligned}\dot{x}_i > 0 &=_{def} Dsign_i = \{1\} \vee Dsign_i = \{-1, 0, 1\}, \\ \dot{x}_i < 0 &=_{def} Dsign_i = \{-1\} \vee Dsign_i = \{-1, 0, 1\}, \\ \dot{x}_i = 0 &=_{def} Dsign_i = \{0\} \vee Dsign_i = \{-1, 0, 1\}.\end{aligned}$$

Notons que le fait que $\dot{x}_i > 0$ et $\dot{x}_i < 0$ ne s'excluent pas découle de l'utilisation d'inclusions différentielles (section 3.2) : lorsque un état satisfait $Dsign_i = \{-1, 0, 1\}$, les trois propositions ci-dessus sont vraies pour cet état.

La conjonction de toutes les propositions atomiques vraies dans un état D est notée inD . Egalement, les propositions *isInstantaneous* et *isPersistent*, définies respectivement comme $Dsign_i = \{0\}$ et $\neg isInstantaneous$, représentent naturellement le caractère instantané ou persistant d'un état. Etant donné que les états instantanés ne sont pas observables en pratique, on pourra ainsi préciser que certaines propriétés observées du système doivent être vérifiées par des états persistants seulement.

$$\begin{aligned}inD &=_{def} \bigwedge \{p \in \Pi \mid D \models_{KS} p\}, \\ isInstantaneous &=_{def} Dsign_i = \{0\}, \\ isPersistent &=_{def} \neg isInstantaneous.\end{aligned}$$

La proposition *isSteady* est définie comme la conjonction des propriétés $\dot{x}_i = 0$:

$$isSteady =_{def} \bigwedge_{i \in \{1, \dots, n\}} \dot{x}_i = 0.$$

Un état satisfaisant cette proposition est naturellement appelé un *état d'équilibre*. Une propriété essentielle de ces états d'équilibres est que tous les points \mathbf{x} du domaine de flot correspondant satisfont $\mathbf{0} \in H(\mathbf{x})$, et sont donc tous des points d'équilibres du système LPM, au sens où en chaque point \mathbf{x} , il existe une solution $\boldsymbol{\xi}$ satisfaisant $\boldsymbol{\xi}(t) = \mathbf{x}$, pour tout $t \geq 0$. La présence d'états d'équilibre, et le cas échéant, leur nombre, donne des informations précieuses sur la dynamique du système. Le système de transitions LPM qualitatif représenté figure 4.10(a) contient trois états d'équilibres, $D^{4.2}$, $D^{7.1}$ et $D^{11.3}$.

Pour illustrer ces notions, considérons l'état $D^{11.2}$, représenté figure 4.8. Cet état satisfait, par exemple, les propositions atomiques $x_a < \theta_a^1$, $x_b < \kappa_b/\gamma_b$ et $Dsign_a = \{-1\}$, et les propriétés $\dot{x}_a < 0$ et *isPersistent*, mais pas *isSteady*. Egalement, $inD^{11.2}$ représente l'expression $x_a \geq 0 \wedge x_a < \theta_a^1 \wedge x_b > \theta_b \wedge x_b < \kappa_b/\gamma_b \wedge Dsign_a = \{-1\} \wedge Dsign_b = \{1\}$, dans laquelle ont été omis les termes que l'on peut déduire directement des contraintes d'inégalité sur les paramètres (4.3)-(4.4), tels que $x_a < \theta_a^2$, $x_a \leq \theta_a^2$, $x_a < max_a$ et $x_a \leq max_a$, sachant que $D^{1.1}$ satisfait $x_a < \theta_a^1$.

Nous avons vu ci-dessus, qu'à partir des propositions atomiques et les opérateurs logiques classiques, il est possible d'exprimer des propriétés portant sur l'état du système. Nous allons voir dans la suite, qu'en utilisant les opérateurs spécifiques aux logiques temporelles, il est également possible d'exprimer des propriétés portant sur les comportements du système.

Propriétés sur les comportements

Les propriétés que nous présentons dans cette section peuvent être exprimées en CTL ou en μ -calcul. Nous avons fait le choix de les exprimer en CTL, car la compréhension des formules de CTL est en général plus aisée.

Une des propriétés les plus fréquemment utilisée concerne l'atteignabilité d'un état : « le système peut atteindre un état satisfaisant une propriété p ». On traduit cette propriété en disant qu'il existe une exécution telle qu'un certain état futur satisfait la propriété p . Cette propriété s'exprime très simplement en CTL : $\mathbf{EF}p$. On peut ainsi, par exemple, exprimer que le système peut passer par un état dans lequel la concentration de la protéine A augmente

$$\mathbf{EF}(\dot{x}_a > 0),$$

ou que le système peut atteindre un état d'équilibre dans lequel la concentration de la protéine B est nulle, ou en pratique, non détectable :

$$\phi_1 = \mathbf{EF}(isSteady \wedge x_b = 0).$$

Une propriété d'atteignabilité plus complexe, mais fréquemment rencontrée lorsque l'on traduit des données obtenues par des techniques à faible taux d'échantillonnage, notamment par des puces à ADN, est que « le système peut passer, dans un ordre donné, par un ensemble d'états D^i satisfaisant une propriété p_i ». Cette propriété peut s'exprimer par des formules du type $\mathbf{EF}(p_1 \wedge \mathbf{EF}(p_2 \wedge \mathbf{EF}(\dots)))$. Supposons par exemple, que l'on observe, pour le réseau d'inhibition réciproque, que le système peut passer successivement par un état où la concentration de la protéine B augmente, puis dans un état où elle diminue, et enfin, atteint un état d'équilibre. Alors, on peut traduire cette observation par la formule suivante :

$$\phi_2 = \mathbf{EF}(\dot{x}_b > 0 \wedge \mathbf{EF}(\dot{x}_b < 0 \wedge \mathbf{EF} isSteady))$$

Si par contre, les données sont obtenues par des techniques à fort taux d'échantillonnage, notamment par des techniques de gènes rapporteurs, pour lesquelles on peut raisonnablement supposer que l'on ne manque pas d'événements significatifs de la dynamique, alors, on obtient des propriétés de la forme : « le système peut passer, dans un ordre donné, par un ensemble de séquences d'états S^i satisfaisant globalement une propriété p_i ». On exprime ainsi que le système satisfait pendant un certain temps p_1 , puis il satisfait pendant un certain temps p_2 , et ainsi de suite. La formalisation de cette propriété utilise l'opérateur *Until*, $\mathbf{U} : \mathbf{E}p_1 \mathbf{U} (\mathbf{E}p_2 \mathbf{U} (\dots))$. Nous rappelons que la formule $\mathbf{E}p_1 \mathbf{U} p_2$ est vraie pour un état s_0 , si il existe un comportement du système tel que la propriété p_2 est vraie pour un certain état futur s , et tel que la propriété p_1 est vraie pour tous les états entre s_0 et s (voir section 2.2.2). La même observation que précédemment, mais en faisant l'hypothèse que l'on ne manque pas d'événements significatifs de la dynamique, se traduit en l'existence d'un comportement du système le long duquel x_b augmente *continûment* pendant

un certain temps, puis diminue *continûment* jusqu'à ce que le système atteigne un état d'équilibre. Alors, la formule de CTL correspondante est :

$$\phi_3 = \mathbf{E} \dot{x}_b > 0 \mathbf{U} (\mathbf{E} \dot{x}_b < 0 \mathbf{U} isSteady)$$

Tandis que les propriétés précédentes portent sur l'existence d'une exécution particulière (*i.e.* la potentialité), il est aussi possible de spécifier des contraintes qui doivent être satisfaites pour toutes les exécutions (*i.e.* l'inévitabilité). Le premier type de propriété est typiquement rencontré lorsque l'on traduit une observation, étant donné que l'on n'observe jamais qu'un comportement possible du système. Le second type de propriété, qui exprime des propriétés générales sur tous les comportements possibles du système, est plutôt utilisé dans des phases d'analyse des résultats. Il est important de remarquer ici qu'étant donné l'objectif principal de ce travail, la validation de modèle par comparaison entre prédictions et observations, il est essentiel de pouvoir exprimer les propriétés de potentialité dans la logique temporelle choisie. Cela justifie notre choix de nous focaliser sur des logiques arborescentes (comme CTL et le μ -calcul) plutôt que sur des logiques linéaires (comme LTL) qui quantifient implicitement sur toutes les exécutions et qui ne peuvent pas exprimer la potentialité.

Une propriété d'inévitabilité fréquemment utilisée s'exprime sous la forme : « le système atteindra nécessairement un état satisfaisant la propriété p ». Ce type de propriété permet d'identifier des « étapes obligatoires » par lesquelles le système passe nécessairement, par exemple des checkpoints, et par conséquent peut suggérer des cibles privilégiées pour éventuellement modifier le comportement du système. Cette propriété s'exprime de façon générale sous la forme $\mathbf{AF}p$. Par exemple, la propriété

$$\phi_4 = \mathbf{AF}(inD^{4.2} \vee inD^{7.1} \vee inD^{11.3})$$

exprime l'inévitabilité pour le système d'arriver dans un des trois états d'équilibre du système LPM (3.4)-(3.5).

Le dernier type de propriété que nous considérerons sont les propriétés exprimant des invariants : « le système satisfait toujours une propriété p ». Pour notre exemple de réseau d'inhibition réciproque, on peut ainsi formuler une propriété d'exclusion réciproque :

$$\phi_5 = \mathbf{AG}(\neg(x_a > \theta_a^1 \wedge x_b > \theta_b)),$$

exprimant que l'on a jamais les protéines A et B présentes en concentration élevée en même temps.

5.2.2 Propriétés exprimables en μ -calcul seulement

Toutes les propriétés précédentes, formulées en CTL, s'expriment également en μ -calcul, strictement plus expressif que CTL. D'autres propriétés, également biologiquement pertinentes, ne peuvent être exprimées qu'en μ -calcul. Nous en voyons deux exemples dans ce chapitre.

« Après un certain temps, le système satisfait toujours la propriété p ». On peut reformuler cette propriété comme « pour tous les comportements possibles du système, après un certain temps, la propriété p sera toujours vraie », qui s'exprime alors en CTL*, une logique strictement plus expressive que CTL mais qui utilise les mêmes opérateurs, comme $\mathbf{AFG}p$. Cette formule n'est pas dans CTL, et n'admet pas de formule CTL équivalente [53].

En particulier, elle n'est pas équivalente à $\mathbf{AFAG}p$ (figure 5.2(a)). Une propriété que l'on voudrait naturellement tester est qu'après un certain temps, le système reste toujours dans un état persistant. Cette propriété est formalisée en μ -calcul d'alternance 2 par

$$\phi_6 = \mu X_1. \nu X_2. ((isPersistent \vee \Box X_1) \wedge \Box X_2).$$

Les phénomènes oscillatoires jouent un rôle fondamental dans le fonctionnement de nombreux systèmes biologiques [102]. Pouvoir exprimer la présence, potentielle ou inévitable, d'oscillations est donc essentiel. La propriété « le système peut présenter des oscillations » peut être traduite par l'existence d'un comportement dans lequel la propriété « si p est vraie alors elle sera fausse dans un instant futur, et si p est fausse alors elle sera vraie dans un instant futur », est vraie à tout instant, où p représente, par exemple, la présence d'une protéine en concentration supérieure à un certain seuil. Cette propriété est formalisée en CTL* comme $\mathbf{EG}(p \rightarrow \mathbf{F}\neg p \wedge \neg p \rightarrow \mathbf{F}p)$. Elle n'est pas dans CTL, et n'admet pas de formule CTL équivalente (théorème 3 de [53]). En particulier, elle n'est pas équivalente à $\mathbf{EG}(p \rightarrow \mathbf{EF}\neg p \wedge \neg p \rightarrow \mathbf{EF}p)$ (figure 5.2(b)). Elle peut par contre être exprimée en μ -calcul. Par exemple, la propriété de μ -calcul d'alternance 2

$$\begin{aligned} \phi_7 = \nu X. (& (\dot{x}_a > 0 \rightarrow \mu Y_1. ((\dot{x}_a \leq 0 \wedge X) \vee \Diamond Y_1)) \\ & \wedge (\dot{x}_a \leq 0 \rightarrow \mu Y_2. ((\dot{x}_a > 0 \wedge X) \vee \Diamond Y_2))), \end{aligned}$$

exprime l'existence d'oscillations dans la concentration de la protéine A. Nous rencontrons un autre exemple de cette propriété au chapitre 8 lors de l'analyse d'un modèle de la réponse au stress nutritionnel chez *E. coli*.

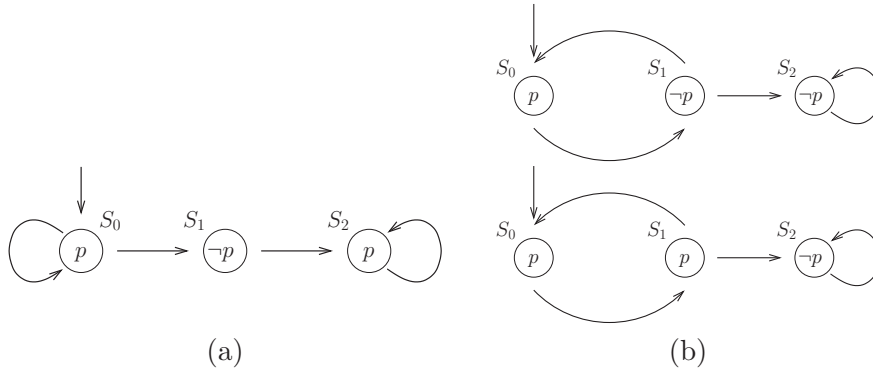


FIG. 5.2 – Propriétés inexprimables en CTL. (a) Les propriétés $\phi_A = \mathbf{AFG}p$ et $\phi_B = \mathbf{AFAG}p$ ne sont pas équivalentes : la structure de Kripke satisfait ϕ_A , mais pas ϕ_B [53]. (b) Les propriétés $\phi_C = \mathbf{EG}(p \rightarrow \mathbf{F}\neg p \wedge \neg p \rightarrow \mathbf{F}p)$ et $\phi_D = \mathbf{EG}(p \rightarrow \mathbf{EF}\neg p \wedge \neg p \rightarrow \mathbf{EF}p)$ ne sont pas équivalentes : contrairement à la formule ϕ_C , la formule ϕ_D ne permet pas de distinguer la structure de Kripke « présentant des oscillations » (haut) de celle n'en présentant pas (bas).

5.3 Vérification de propriétés temporelles

Nous avons vu dans la section 5.1 comment reformuler un système de transitions LPM qualitatif en une structure de Kripke et dans la section 5.2 comment exprimer des propriétés sur la dynamique du système à l'aide de logiques temporelles. Il est maintenant possible

d'utiliser des outils de model checking pour vérifier des propriétés biologiques d'un réseau de régulation génique. Selon la logique employée, CTL ou le μ -calcul, le model checker utilisé est différent. Dans cette section, nous verrons notamment qu'il est important de prendre en compte les spécificités de la méthode d'analyse présentée au chapitre 4 pour obtenir des résultats biologiquement pertinents.

5.3.1 Vérification de propriétés exprimées en CTL

Dans cette section, nous testons si les propriétés formulées en CTL à la section précédente sont satisfaites par un ou plusieurs états initiaux de la structure de Kripke représentant la dynamique du système. Nous utilisons le model checker NuSMV [49, 50].

Dans la table 5.1 sont résumés les résultats de l'évaluation des propriétés de logique temporelle proposées dans la section précédente, sur la structure de Kripke représentée figure 5.1. Les deux premières propriétés, exprimant respectivement que le système peut atteindre un état d'équilibre dans lequel la concentration de la protéine B est nulle (ϕ_1), et que le système peut passer successivement par un état où la concentration de la protéine B augmente, puis dans un état où elle diminue, et enfin, atteindre un état d'équilibre (ϕ_2), sont vérifiées par les prédictions. Un diagnostic est également proposé pour les différentes propriétés testées.

Numéro	Propriété	\mathcal{D}_0	Résultat
ϕ_1	$\mathbf{EF}(isSteady \wedge x_b = 0)$	$\{D^{1.1}\}$	Vrai : $diag_1$
ϕ_2	$\mathbf{EF}(\dot{x}_b > 0 \wedge \mathbf{EF}(\dot{x}_b < 0 \wedge \mathbf{EF} isSteady))$	$\{D^{1.1}\}$	Vrai : $diag_1$
ϕ_3	$\mathbf{E} \dot{x}_b > 0 \mathbf{U} (\mathbf{E} \dot{x}_b < 0 \mathbf{U} isSteady)$	$\{D^{1.1}\}$	Faux : –
ϕ'_3	$\mathbf{E}(\dot{x}_b > 0 \vee isInstantaneous) \mathbf{U} (\mathbf{E}(\dot{x}_b < 0 \vee isInstantaneous) \mathbf{U} isSteady)$	$\{D^{1.1}\}$	Vrai : $diag_1$
ϕ_4	$\mathbf{AF}(inD^{4.2} \vee inD^{7.1} \vee inD^{11.3})$	\mathcal{D}	Faux : $diag_2$

$$diag_1 = (D^{1.1}, D^{2.2}, D^{3.2}, D^{4.2}, D^{4.1}),$$

$$diag_2 = (D^{1.1}, D^{1.1}, \dots),$$

Le symbole – signifie que le diagnostic n'est pas une simple exécution.

TAB. 5.1 – Vérification de propriétés exprimées en CTL, portant sur la dynamique du réseau d'inhibition réciproque présenté figure 3.1. \mathcal{D}_0 indique l'ensemble des états initiaux. Le résultat comprend la valeur de vérité de la formule et un diagnostic (témoin ou contre-exemple).

Présence d'états instantanés et système de transitions LPM qualitatif filtré

Etonnamment, le model checker ne trouve aucun comportement le long duquel x_b augmente continûment pendant un certain temps, puis diminue continûment jusqu'à ce que le système atteigne un état d'équilibre (propriété ϕ_3). Pourtant, l'exécution ($D^{1.1}, D^{2.2}, D^{3.2}, D^{4.2}, D^{4.1}, D^{4.1}, \dots$), représentée dans la figure 4.10(b) semble la vérifier (section 4.2.2). C'est la présence de l'état instantané $D^{2.2}$, pour lequel ni $\dot{x}_b < 0$ ni $\dot{x}_b > 0$ n'est vrai qui conduit le model checker à infirmer cette propriété. La présence d'états instantanés peut être prise en compte en modifiant la propriété, comme montré dans la table 5.1 (propriété ϕ'_3). Une autre possibilité est, étant donné la faible pertinence biologique de ces états, de travailler directement avec le système de transitions LPM qualitatif *filtré*, c'est-à-dire, dans lequel ces états ont été éliminés (et les connexions entre prédécesseurs et successeurs

restaurées). La structure de Kripke filtrée, associée au système de transitions LPM qualitatif Σ -QTS pour l'exemple du réseau d'inhibition réciproque, est représentée figure 5.3. Pour le système filtré, la propriété ϕ_3 est vraie.

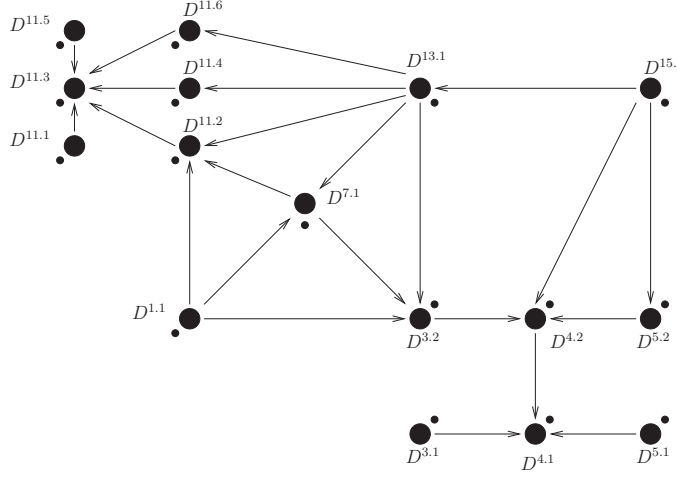


FIG. 5.3 – Structure de Kripke filtrée (*i.e.* dans laquelle les états instantanés ont été éliminés), associée au système de transitions LPM qualitatif Σ -QTS et obtenue à partir de la structure de Kripke Σ -KS représentée figure 5.1. Les symboles \bullet représentent des boucles sur les états.

Présence de boucles et contraintes d'équité

Dans la table 5.1, la dernière formule, ϕ_4 , exprime l'inévitabilité pour le système d'arriver dans un des trois états d'équilibre, $D^{4.1}$, $D^{7.1}$ ou $D^{11.3}$. Il peut être montré que, pour tous les paramètres vérifiant les contraintes d'inégalité (4.3)-(4.4), toutes les solutions du système LPM (3.4)-(3.5) atteignent ou convergent asymptotiquement vers un des trois points d'équilibre. Nous nous attendons donc à ce que la propriété ϕ_4 soit vérifiée par le système de transitions LPM qualitatif correspondant. Or cela n'est pas le cas. Le model checker donne comme contre-exemple l'exécution $(D^{1.1}, D^{1.1}, \dots)$. Cette exécution, dans laquelle le système reste toujours dans $D^{1.1}$, en utilisant infiniment souvent la boucle sur $D^{1.1}$, ne correspond à aucune solution du système LPM sur l'intervalle de temps $[0, +\infty]$.

Pour exclure ces comportements artificiels, nous utilisons des contraintes d'équité (section 2.2.2). Rappelons que la notion d'équité a justement été introduite pour exprimer l'idée que certains processus doivent progresser[13, 90]. En ajoutant la contrainte d'équité *isSteady*, on restreint l'ensemble des exécutions considérées aux seules exécutions, dites équitables, qui vérifient infiniment souvent cette propriété. L'ajout de contraintes d'équité nous permet donc d'exclure certaines exécutions, qui, comme $(D^{1.1}, D^{1.1}, \dots)$, ne correspondent à aucun comportement réel du système original. En utilisant la contrainte d'équité *isSteady*, nous obtenons alors le résultat attendu (table 5.2).

Le résultat de l'évaluation, avec contraintes d'équité, de la propriété traduisant l'exclusion mutuelle est donné dans la table 5.2 (propriété ϕ_5). Comme attendu, la propriété d'exclusion mutuelle est vraie.

Numéro	Propriété	\mathcal{D}_0	Contrainte d'équité	Résultat
ϕ_4	$\mathbf{AF}(inD^{4.2} \vee inD^{7.1} \vee inD^{11.3})$	\mathcal{D}	<i>isSteady</i>	Vrai : –
ϕ_5	$\mathbf{AG}(\neg(x_a > \theta_a^1 \wedge x_b > \theta_b))$	$\{D^{1.1}\}$	<i>isSteady</i>	Vrai : –

TAB. 5.2 – Vérification de propriétés exprimées en CTL, avec contrainte d'équité, portant sur la dynamique du réseau d'inhibition réciproque présenté figure 3.1. \mathcal{D}_0 indique l'ensemble des états initiaux.

Abstraction et interprétation des résultats

Nous avons vu, à la section 4.2.2, que le système de transitions LPM qualitatif Σ -QTS obtenu par abstraction qualitative est une approximation conservative du système LPM original Σ . Or, par model checking, nous vérifions les propriétés de Σ -QTS (ou plus précisément de Σ -KS). Est-ce que les conclusions obtenues pour Σ -QTS sont aussi valables pour Σ ? D'après les résultats présentés à la section 2.2.3 sur la vérification de systèmes abstraits, nous savons que si une propriété dite universelle est vérifiée par Σ -QTS, alors elle l'est nécessairement par Σ -TS, et ainsi, qu'une propriété équivalente est vraie pour le système original Σ (théorème 3). Rappelons que les propriétés universelles correspondent à des propriétés qui doivent être vérifiées par toutes les exécutions possibles du système. Ces propriétés font partie de \forall CTL ou du μ -calcul universel, pour les logiques que nous considérons. Parmi les résultats obtenus, les résultats de l'évaluation sur Σ -QTS des propriétés ϕ_3 , ϕ_4 et ϕ_5 avec équité, sont également valables pour Σ , car les propriétés $\neg\phi^3$, ϕ^4 et ϕ^5 sont vraies et sont dans \forall CTL.

La notion de propriété universelle est importante pour la validation de modèles. Effectivement, l'observation expérimentale d'un comportement se traduit le plus souvent par une propriété, ϕ , portant sur l'existence d'une exécution satisfaisant des propriétés particulières (voir les propriétés ϕ_1 , ϕ_2 et ϕ_3). La négation de cette propriété, $\neg\phi$, est alors une propriété universelle. Ainsi, si la propriété ϕ n'est pas satisfaite par le système abstrait, Σ -QTS, $\neg\phi$ est alors vraie, et le théorème 3 s'applique, ce qui nous permet de prouver ainsi qu'aucune solution du système original n'est cohérente avec l'observation. La contradiction étant formellement démontrée, le modèle peut donc être rejeté. Dans les autres cas, le résultat obtenu sur le système abstrait, Σ -QTS, peut être dû à l'existence d'exécutions artéfactuelles (*i.e.* qui sont dans Σ -QTS, mais pas dans Σ -TS), et il n'est pas possible d'en tirer des conclusions pour le système original, Σ .

5.3.2 Vérification de propriétés exprimées en μ -calcul propositionnel

Certaines propriétés biologiquement intéressantes, comme la présence d'oscillations, ne peuvent être exprimées en CTL, mais peuvent l'être en μ -calcul (section 5.2.2). Nous utiliserons la boîte à outils CADP pour vérifier ces propriétés. Les outils de CADP utilisent un formalisme basé sur les actions (section 2.2.2). Notre problème étant posé dans un formalisme basé sur les états, il est nécessaire de pouvoir passer d'un formalisme vers l'autre. Concrètement, cela implique de transposer le problème de la vérification d'une formule de μ -calcul propositionnel sur une structure de Kripke en un problème équivalent de vérification d'une formule de μ -calcul modal sur un système de transitions étiquetées. Dans [157], Mateescu propose une méthode simple pour réaliser cette transposition. Dans cette section, nous n'en exposons que le principe. Également, nous montrons comment en pratique cette méthode est utilisée pour tester si des propriétés exprimées en μ -calcul sont

vérifiées par les prédictions obtenues sur la dynamique du système.

Dans [157], Mateescu définit deux applications, W et t , transformant respectivement une structure de Kripke, KS , et une formule de μ -calcul propositionnel, ϕ , en un système de transitions étiquetées et une formule de μ -calcul modal, préservant la relation de satisfaction, c'est-à-dire telles que :

$$KS \models \phi \text{ si et seulement si } W(KS) \models t(\phi) \quad (\text{théorème 2 de [157]})$$

Etant donné la structure de Kripke $\Sigma\text{-KS} = (\mathcal{D}, \rightarrow_{KS}, \models_{KS})$, W consiste simplement à étiqueter les transitions du graphe $G = (\mathcal{D}, \rightarrow_{KS})$ de telle sorte que l'on ait $D \xrightarrow{L(D)} D'$, où $L : \mathcal{D} \rightarrow 2^{AP}$ est la fonction associant à tout état l'ensemble des propositions atomiques vraies dans cet état : $L(D) = \{p \in AP \mid D \models_{KS} p\}$. La transformation t consiste à remplacer syntaxiquement dans les formules de μ -calcul propositionnel, p , $\diamond\phi$, et $\Box\phi$, par $\langle p \rangle \text{True}$, $\langle - \rangle \phi$, et $[-]\phi$, respectivement, où le symbole $-$ correspond à une étiquette quelconque. Il est alors possible d'utiliser CADP pour vérifier des propriétés complexes, formulées en μ -calcul, sur la dynamique du système. Nous pouvons par exemple tester si les propriétés sur la dynamique du réseau d'inhibition mutuelle, énoncées à la section 5.2.2, sont satisfaites par les prédictions obtenues à partir du système LPM (3.4)-(3.5) et des contraintes d'inégalité sur les paramètres (4.3)-(4.4).

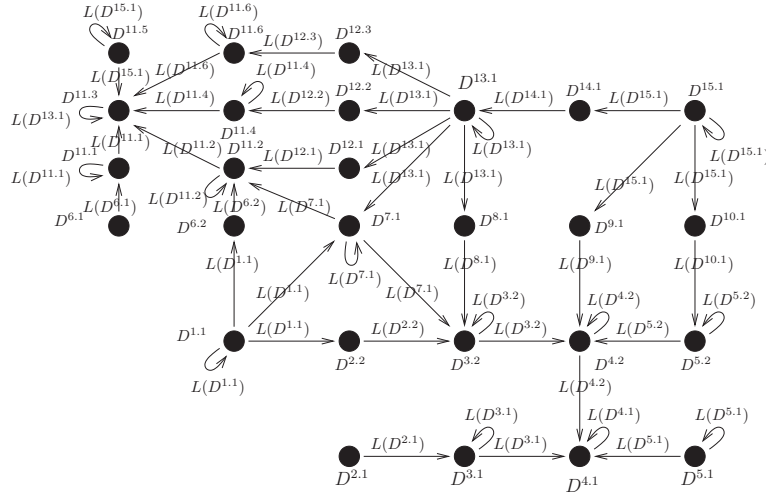


FIG. 5.4 – Traduction de la structure de Kripke $\Sigma\text{-KS}$ représentée figure 5.1 (voir aussi figure 4.10) en système de transitions étiquetées, $W(\Sigma\text{-KS})$, comme proposé dans [157].

Dans la figure 5.4, est représenté le système de transitions étiquetées, $W(\Sigma\text{-KS})$, correspondant à la structure de Kripke représentée figure 5.1 (voir également figure 4.10). La table 5.3 donne la traduction en μ -calcul modal des propriétés formulées en μ -calcul propositionnel à la section 5.2.2. Nous rappelons que ces propriétés expriment respectivement qu'après un certain temps, le système reste toujours dans un état persistant (propriété ϕ_6) et que le système peut présenter des oscillations pour la concentration de la protéine A (propriété ϕ_7). Egalement, pour comparaison, nous avons ajouté dans cette table une formule traduisant la propriété ϕ_1 , et équivalente à la formule de CTL $\mathbf{EF}(isSteady \wedge x_b = 0)$ (voir section 5.2.1).

$$\begin{aligned}
t(\phi_1) &= \mu X.((\langle isSteady \wedge x_b = 0 \rangle \text{True}) \vee \langle - \rangle X), \\
t(\phi_6) &= \mu X.\nu Y.((\langle isPersistent \rangle \text{True} \vee [-]X) \wedge [-]Y), \\
t(\phi_7) &= \nu X.(\langle \dot{x}_a > 0 \rangle \text{True} \rightarrow \mu Y_1.(\langle \dot{x}_a \leq 0 \rangle \text{True} \wedge X) \vee \langle - \rangle Y_1) \\
&\quad \wedge (\langle \dot{x}_a \leq 0 \rangle \text{True} \rightarrow \mu Y_2.(\langle \dot{x}_a \leq 0 \rangle \text{True} \wedge X) \vee \langle - \rangle Y_2)).
\end{aligned}$$

TAB. 5.3 – Traduction en μ -calcul modal de quelques formules exprimées en μ -calcul propositionnel, comme proposé par Mateescu [157].

Le résultat de l'évaluation de ces propriétés à l'aide des outils de CADP (Evaluator et XTL) est donné dans la table 5.4. Premièrement, nous retrouvons le résultat, obtenu dans la section précédente, que la propriété ϕ_1 est vraie. Egalement, la seconde propriété testée (ϕ_6) est vraie. Le fait qu'après un certain temps, le système reste toujours dans un état persistant, est en réalité une conséquence de l'absence de cycles (*i.e.* de composantes fortement connexes autres que des boucles) dans la structure de Kripke Σ -KS. L'absence d'oscillations pour la concentration de la protéine A (propriété ϕ_7), est également une conséquence de l'absence de cycles dans Σ -KS.

Numéro	Propriété	\mathcal{D}_0	Résultat
ϕ_1	$\mu X.((isSteady \wedge x_b = 0) \vee \diamond X)$	$\{D^{1.1}\}$	Vrai
ϕ_6	$\mu X.\nu Y.((isPersistent \vee \square X) \wedge \square Y)$	\mathcal{D}	Vrai
ϕ_7	$\nu X.(\langle \dot{x}_a > 0 \rangle \text{True} \rightarrow \mu Y_1.(\langle \dot{x}_a \leq 0 \rangle \text{True} \wedge X) \vee \diamond Y_1) \\ \wedge (\langle \dot{x}_a \leq 0 \rangle \text{True} \rightarrow \mu Y_2.(\langle \dot{x}_a > 0 \rangle \text{True} \wedge X) \vee \diamond Y_2))$	\mathcal{D}	Faux

TAB. 5.4 – Vérification de propriétés exprimées en μ -calcul propositionnel, portant sur la dynamique du réseau d'inhibition réciproque. \mathcal{D}_0 indique l'ensemble des états initiaux.

Les remarques faites à la section précédente, sur les conséquences de la présence d'états instantanés et de boucles sur certains états, ainsi que sur le fait que Σ -KS ne soit qu'une approximation conservative de Σ , sont également valables pour la vérification de propriétés exprimées en μ -calcul. Les solutions proposées, à savoir, utiliser un système « filtré », utiliser la notion d'équité et distinguer les propriétés dites universelles, *i.e.* du μ -calcul universel dans ce cas (section 2.2.3), s'appliquent de façon tout à fait similaire.

Dans ce chapitre, nous avons proposé une reformulation simple du système de transitions LPM qualitatif, Σ -QTS, en structure de Kripke, Σ -KS. Egalement, nous avons montré comment un large éventail de propriétés biologiquement pertinentes pouvaient être exprimées en logique temporelle. La vérification des propriétés peut alors être réalisée de façon automatique. Une application sur l'exemple simple du réseau d'inhibition réciproque présenté à la section 3.1.1 nous a permis d'illustrer l'importance de prendre en compte certaines spécificités de la méthode d'analyse présentée au chapitre 4 pour obtenir des résultats de vérification biologiquement pertinents.

Nous verrons dans la partie Applications, que par cette méthode, il est effectivement possible d'analyser efficacement les grands systèmes de transitions obtenus pour des modèles biologiquement intéressants. Mais auparavant, nous présentons dans la partie suivante l'implémentation, dans une nouvelle version de l'outil Genetic Network Analyzer, de la méthode présentée dans les chapitres 3, 4 et 5.

Troisième partie

Outil informatique

6 Genetic Network Analyzer 6.0

La méthode d'analyse symbolique et de vérification de modèles LPM qualitatifs développée dans la partie précédente, est implémentée dans une nouvelle version de l'outil informatique *Genetic Network Analyzer*, GNA 6.0.

GNA 6.0 a été développé à partir d'une version précédente de l'outil (GNA 5.5). GNA 5.5 implémente la méthode proposée par de Jong *et al.* [70] et a été développé par Michel Page et Hidde de Jong [68]. Les deux versions diffèrent essentiellement sur deux points. Premièrement, GNA 6.0 utilise une partition en domaines de flot, tandis que la version précédente utilise une partition en domaines de mode. L'utilisation d'une description plus fine de l'état du système a nécessité de réécrire substantiellement le coeur de l'outil, réalisant l'analyse qualitative. L'architecture globale du logiciel a néanmoins été préservée. Deuxièmement, GNA 6.0 intègre des fonctionnalités d'export des résultats de l'analyse (graphes de transitions) vers deux model checkers, CADP et NuSMV. Le parseur pour la lecture des modèles et l'interface graphique ont également dû être adaptés.

GNA 6.0 est librement disponible pour la recherche académique¹. De plus, le logiciel fait actuellement l'objet d'un transfert technologique pour sa diffusion gratuite auprès d'utilisateurs académiques et sa commercialisation auprès d'utilisateurs non académiques par la société Genostar Technologies.

Dans ce chapitre, nous décrivons GNA 6.0 selon deux points de vue complémentaires. Tout d'abord, nous adopterons le point de vue utilisateur et nous présenterons les principales fonctionnalités de l'outil. Puis, nous adopterons le point de vue informatique et nous présenterons quelques aspects liés à l'architecture du logiciel et aux algorithmes utilisés.

6.1 Utilisation de GNA 6.0

6.1.1 Généralités

GNA 6.0 est écrit en Java 1.4. Il fonctionne donc sur toute plateforme munie d'une machine virtuelle Java, version 1.4 ou supérieure. Notamment, GNA 6.0 a été testé sur les plateformes Linux (Fedora 2 et Redhat 9), Windows (NT 4.0, 2000 et XP), et Macintosh (OS X). Le code a été déposé à l'Agence pour la Protection des Programmes (référencement en cours).

La distribution consiste en une archive d'environ 7 Mo, incluant une documentation au format HTML ainsi qu'un ensemble de modèles et de conditions initiales, avec notamment les modèles et conditions initiales présentés dans la partie IV concernant l'initiation de la sporulation chez *B. subtilis* (chapitre 7) et la réponse au stress nutritionnel chez *E. coli* (chapitre 8).

¹Contact : hidde.de-jong@inrialpes.fr.

6.1.2 Principales fonctionnalités

Utilisation via des commandes en ligne ou via l'interface graphique VisualGNA

GNA peut être utilisé soit via des commandes en ligne soit via une interface graphique appelée VisualGNA. L'utilisation par commande en ligne permet, étant donné un modèle et éventuellement des conditions initiales, de calculer le graphe de transitions correspondant et de l'exporter vers les model checkers NuSMV et CADP. La syntaxe est

```
gna modelFile initCondFile presentMode traceLvl sampleLimit nusmvExportFile cadpExportFile,
```

où `presentMode` et `traceLvl` déterminent la quantité d'information affichée lors du calcul, `sampleLimit` permet éventuellement de limiter l'analyse à un certain nombre maximal d'états, et les autres arguments correspondent à des noms de fichiers avec une signification évidente. La donnée des fichiers `initCondFile`, `nusmvExportFile` ou `cadpExportFile` est optionnelle.

L'utilisation par commandes en ligne permet l'analyse systématique d'un grand nombre de modèles, pour un grand nombre de conditions initiales, via l'écriture de scripts. Toutefois, l'utilisation via l'interface graphique est plus conviviale et offre de nombreuses fonctionnalités supplémentaires. Dans la suite, nous nous focalisons sur cette dernière.

Edition de modèles

Via VisualGNA, il est possible de créer, effacer et renommer les variables d'état et d'entrée et les paramètres de seuil, de production et de dégradation. Il est également possible d'éditer les équations d'état et de spécifier les contraintes d'inégalité sur les paramètres. Les modèles ainsi créés peuvent être sauvegardés sous forme de fichiers texte (format `.gna`, voir la documentation de GNA).

Le modèle présenté dans la partie II pour le réseau d'inhibition réciproque et constitué des équations (3.4)-(3.5) et des contraintes d'inégalité (4.3)-(4.4), est représenté dans la figure 6.1. La variable `xa` est complètement spécifiée. Elle est associée à deux seuils, `t_xa1` et `t_xa2`, en plus des paramètres `z_xa` et `max_xa` représentant les concentrations minimale et maximale de la protéine A, à un paramètre de production `k_xa` et à un paramètre de dégradation `g_xa`. L'équation différentielle correspond à (3.4), et le bouton `checkEquation` permet d'en vérifier la syntaxe. Enfin, l'ordre des concentrations de seuil et focales correspond à l'ordre spécifié par les contraintes d'inégalité sur les paramètres (4.3). L'ordre des éléments de cette liste peut être modifié à l'aide des boutons "▲" et "▼". La représentation textuelle du modèle complètement spécifié (fichier `.gna`) est donnée dans l'annexe B.1. Par exemple, la variable `xa`, correspondant à la protéine A, est définie par

```
state-variable: xa
  zero-parameter: z_xa
  box-parameter: max_xa
  threshold-parameters: t_xa1, t_xa2
  synthesis-parameters: k_xa
  degradation-parameters: g_xa
  state-equation:
    d/dt xa = k_xa * s-(xa, t_xa2) * s-(xb,t_xb) - g_xa * xa
  parameter-inequalities:
    z_xa < t_xa1 < t_xa2 < k_xa/g_xa < max_xa ;
```

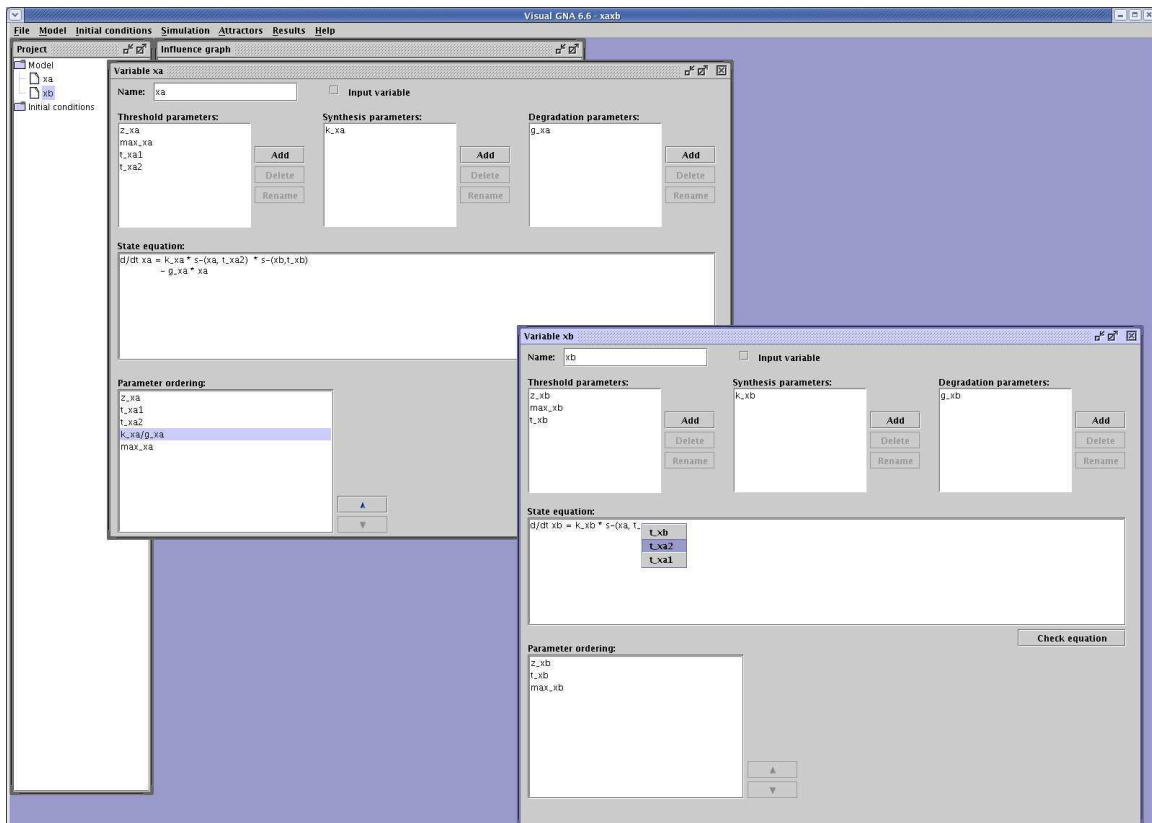


FIG. 6.1 – Capture d'écran de GNA 6.0 montrant le modèle du réseau d'inhibition réciproque présenté partie II en cours d'édition.

Il est à noter que les contraintes d'inégalité sur les paramètres doivent définir un ordre *total* sur les concentrations de seuil et focales. Ceci n'était pas nécessairement le cas avec les modèles acceptés par la version précédente de GNA, pour lesquels un ordre relatif seulement était demandé [70]. Pour les modèles précédemment développés (*i.e.* avec la version 5.5 de GNA), il est possible que l'on ne puisse déduire un ordre total sur les concentrations de seuil et focales, à partir des ordres partiels donnés. Pour cette raison, la compatibilité ascendante des modèles n'est pas totale entre les deux versions de GNA.

Edition de conditions initiales

La concentration initiale des protéines est spécifiée sous la forme d'un intervalle, par la donnée d'une borne inférieure et d'une borne supérieure, éventuellement égales. Ces bornes correspondent soit à des concentrations de seuil, soit à des concentrations focales. Les conditions initiales peuvent être sauvegardées sous forme de fichiers textes (format `.dat`, voir la documentation de GNA). Ensemble, ces contraintes définissent une région hyperrectangulaire de l'espace des phases, correspondant à un ou plusieurs domaines, ou états, initiaux.

Pour illustration, des conditions initiales pour le modèle représenté figure 6.1, sont données figure 6.2. Ces conditions spécifient que les concentrations initiales des deux protéines sont basses (en dessous de leur premier seuil). La représentation textuelle de ces conditions initiales (fichier `.dat`) est donnée dans l'annexe B.1. Par exemple, les conditions initiales pour la protéine A sont définies par

```
xa >= z_xa; xa < t_xa1;
```

Il est à noter que dans la version précédente de GNA, seules les concentrations de seuil pouvaient être utilisées pour spécifier des conditions initiales : il n'était pas possible de spécifier des régions plus petites qu'un domaine de mode. Avec la version 6.0, il est possible de spécifier plus précisément des conditions initiales, ce qui est important étant donné qu'un domaine de mode peut contenir un grand nombre de domaines de flot (par exemple, un domaine de mode régulier de dimension élevée contenant un point d'équilibre).

Analyse de l'espace des phases ou d'atteignabilité

GNA permet soit de calculer à partir d'un modèle, le système de transitions LPM qualitatif entier, soit de calculer à partir d'un modèle et de conditions initiales (*i.e.* d'un ensemble d'états initiaux), le fragment du système de transitions composé seulement des états atteignables à partir des états initiaux. Ces deux types d'analyse correspondent respectivement à une *analyse de l'espace des phases* et à une *analyse d'atteignabilité*, encore appelée *simulation qualitative*. Dans les deux cas, le résultat de l'analyse, *i.e.* le système de transitions LPM qualitatif entier ou partiel, est simplement appelé *graphe de transitions*.

Notons que l'expérience montre que, pour des modèles biologiques, les analyses d'atteignabilité ont en général des propriétés de mise à l'échelle très largement supérieures à l'analyse de l'espace des phases.

Contrôle de la simulation

Une fois qu'un modèle et des conditions initiales sont spécifiés, le graphe de transitions peut être calculé à l'aide de GNA. L'utilisateur peut spécifier un temps de calcul maximal

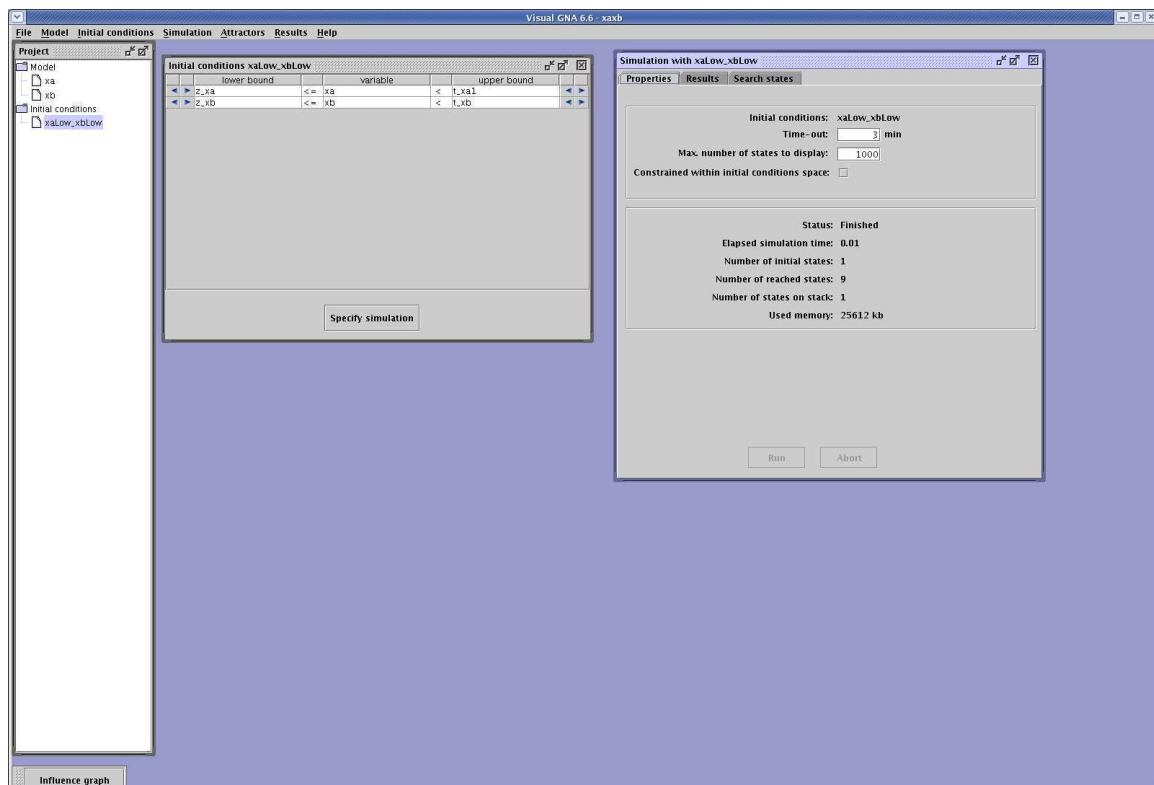


FIG. 6.2 – Capture d'écran de GNA 6.0 montrant des conditions initiales pour le modèle représenté figure 6.1 (au centre). La capture d'écran fait également apparaître des informations sur le déroulement d'une simulation (à droite).

et un nombre maximal d'états à afficher, l'affichage de très grands graphes pouvant être peu informatif et consommateur de ressources. Diverses informations, dont notamment le nombre d'états atteints, le temps écoulé et la quantité de mémoire utilisée, permettent de suivre l'évolution de la simulation. Celle-ci peut être interrompue à tout moment. Dans la figure 6.2, la fenêtre permettant de contrôler la simulation est visible (à droite).

Analyse des résultats de simulation

Différentes possibilités sont offertes par GNA pour l'analyse des résultats de simulation. Premièrement, l'utilisateur dispose d'un ensemble de propriétés sur le graphe de transitions obtenu, notamment le nombre d'états, le nombre d'états initiaux, d'états persistants, d'états d'équilibre, et de composantes fortement connexes, aussi appelées cycles qualitatifs. Également, les états d'équilibre, les composantes fortement connexes et leur bassins d'attraction sont identifiés. Deuxièmement, le graphe de transitions peut être analysé. Les états instantanés, persistants et d'équilibre sont aisément identifiables d'après leur apparence graphique. Les informations concernant les états, *i.e.* les concentrations des protéines et leur sens de variation, sont affichées lorsque le curseur est placé au-dessus d'un état. Il est également possible d'afficher séparément l'évolution temporelle des concentrations des protéines et de leur sens de variation le long d'un chemin sélectionné.

Dans la figure 6.3, sont présentés des résultats de la simulation du comportement du réseau d'inhibition réciproque pour les conditions initiales données figure 6.2. L'évolution du système le long du chemin (S1, S3, S6, S7) est représentée en détail. Ce chemin correspond au chemin ($D^{1.1}$, $D^{6.2}$, $D^{11.2}$, $D^{11.3}$) représenté figure 4.10(b).

Par rapport à la version précédente de GNA, il est important de garder à l'esprit que les états dans le graphe de transitions correspondent dans l'espace des phases à des domaines de flot, et non plus à des domaines de mode. Cette subdivision plus fine de l'espace des phases, ainsi que la donnée du sens de variation des différentes protéines dans les états persistants, apparaît clairement dans le chemin représenté figure 6.3.

D'autres opérations sur le graphe sont possibles comme zoomer, éliminer (*i.e.* « filtrer ») les états instantanés ou sélectionner des sous-graphes pour les analyser séparément. Enfin, il est aussi possible de les exporter pour les analyser par model checking, comme proposé dans le chapitre 5.

Export vers des model checkers

Lors des exports, une description textuelle du graphe et des propriétés des états est générée dans un format reconnu par les model checkers utilisés dans cette thèse, NuSMV [49] et CADP [91]. Rappelons que formellement, le graphe doit être exporté sous la forme d'une structure de Kripke pour l'analyse des propriétés en CTL par NuSMV, alors qu'il doit être exporté sous la forme d'un système de transitions étiquetées pour l'analyse des propriétés en μ -calcul par les outils de CADP (voir les sections 5.3.1 et 5.3.2).

Les exports vers NuSMV sont réalisés dans le langage d'entrée de NuSMV (format `.smv`), spécifié dans [45]. La syntaxe des exports est aisément illustrée à partir d'un exemple. Le résultat de l'export du graphe représenté figure 6.3, ou plus précisément, de la structure de Kripke associée, est présentée dans l'annexe B.2. Par exemple, l'état S1 est défini par

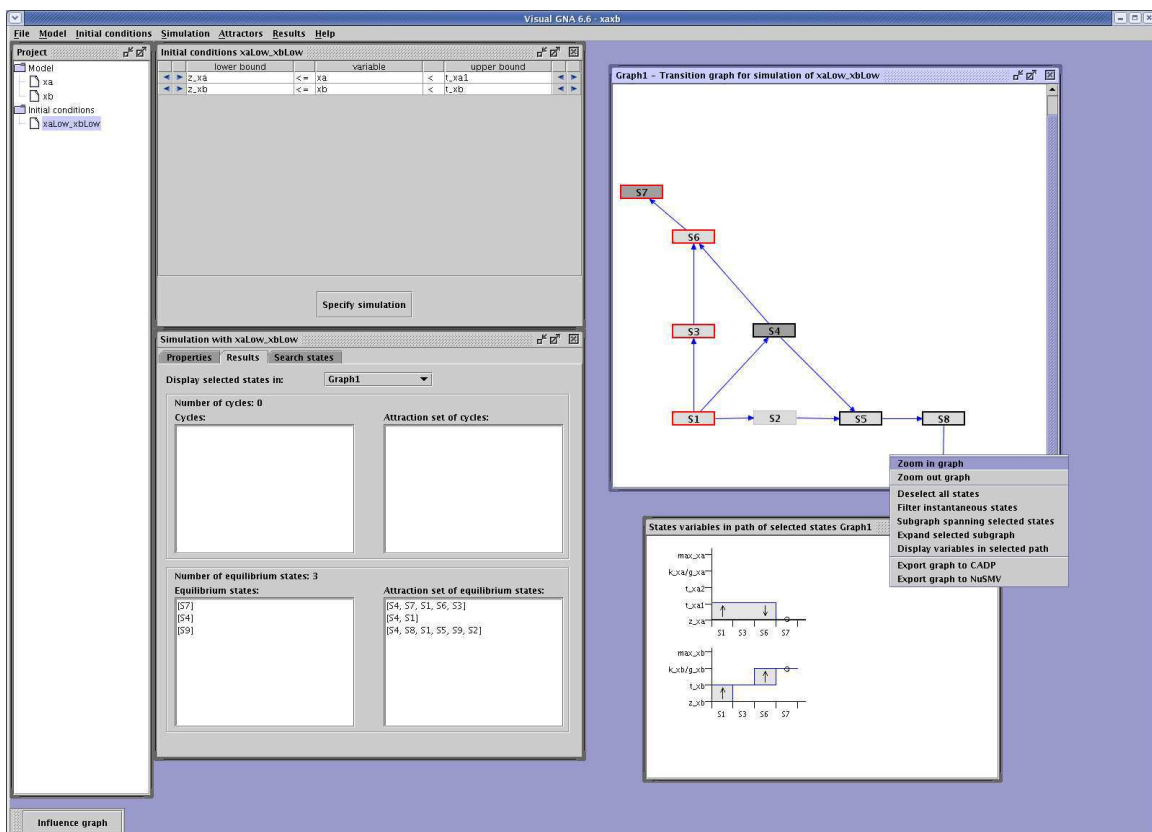


FIG. 6.3 – Capture d’écran de GNA 6.0 montrant le résultat d’une simulation : graphe de transitions, information sur les attracteurs du graphe et analyse détaillée de l’évolution du système le long d’un chemin.

```

inQS1 := QState_number = 1
  & min_xa = GE_z_xa
  & max_xa = LT_t_xa1
  & min_xb = GE_z_xb
  & max_xb = LT_t_xb
  & DSignOfxa= _INC_
  & DSignOfxb= _INC_;

```

où il faut interpréter $\text{min_xa} = \text{GE_z_xa}$, comme $\text{xa} \geq \text{z_xa}$ (GE pour « greater than or equal »), $\text{max_xa} = \text{LT_t_xa1}$, comme $\text{xa} < \text{t_xa1}$ (LT pour « less than »), et également $\text{DSignOfxa} = \text{_INC_}$, comme $d\text{xa}/d\text{t} > 0$. L'interprétation des autres propositions atomiques est très similaire.

L'ensemble des états initiaux est donné par INIT, et la relation de transition, par TRANS, à l'aide de l'opérateur next (voir l'annexe B.2). Ainsi,

```
QState_number = 1 : next(inQS1 | inQS2 | inQS3 | inQS4);
```

signifie que S1 possède quatre successeurs : lui-même, S2, S3 et S4. Certains propositions très utiles sont également systématiquement définies, comme `isInitialState`, `isSteadyState` ou `isInSCCi`, pour décrire l'appartenance aux états initiaux, aux états d'équilibre ou à la *i*ème composante fortement connexe, respectivement.

Les exports vers CADP sont donnés dans le format `.aut` de CADP. A nouveau, nous illustrons la syntaxe à partir d'un exemple. Le résultat de l'export du graphe de transitions obtenu à la section précédente pour le réseau d'inhibition réciproque (figure 6.3), est donné dans l'annexe B.3. Formellement, le fichier d'export représente le système de transitions étiquetées $W(\Sigma\text{-KS})$, où $\Sigma\text{-KS}$ est la structure de Kripke définie à la section 5.1 et W est l'application introduite à la section 5.3.2 transformant une structure de Kripke en un système de transitions étiquetées préservant les propriétés du μ -calcul (au sens défini à la section 5.3.2).

Le fichier d'export consiste en l'énumération de toutes les transitions possibles. Plus précisément, toutes les transitions sont de la forme $(S_i, "L(S_i)", \text{Succ}S_i)$, où S_i est un état, $L(S_i)$, sa description et $\text{Succ}S_i$, un de ses successeurs. Ainsi,

```
(1, "S1: xa:[z_xa,t_xa1[,inc; xb:[z_xb,t_xb[,inc; initial ", 2)
```

signifie que l'état 1 du système de transitions étiquetées est S1, que S1 vérifie les propriétés $\text{xa} \in [z_xa, t_xa1[$, $d\text{xa}/d\text{t} > 0$, $\text{xb} \in [z_xb, t_xb[$, $d\text{xb}/d\text{t} > 0$, que S1 est un état initial, et qu'il a pour successeur l'état 2 du système de transitions étiquetées (qui est S2). Comme précédemment, en plus de l'information correspondant aux propositions atomiques, un certain nombre de propriétés est ajouté systématiquement lors de l'export comme l'appartenance aux états initiaux, aux états d'équilibre ou à des composantes fortement connexes.

La spécification et la vérification des propriétés dynamiques du réseau se fait alors dans l'environnement des différent model checkers.

6.2 Implémentation de GNA 6.0

Dans cette section, nous décrivons l'architecture globale du logiciel et nous présentons quelques aspects algorithmiques importants pour l'efficacité des calculs réalisés par GNA.

6.2.1 Architecture du logiciel

Comme dit précédemment, GNA 6.0 a été développé en Java 1.4 à partir d’une version précédente de l’outil (GNA 5.5) [68]. Le logiciel est composé de 121 classes, totalisant 17800 lignes de code dont environ 4500 de commentaires. Les modifications entre la version 6.0 du logiciel et la précédente sont très nombreuses. Par exemple, près de 35% des lignes ont été modifiées (60% dans le kernel), 6 classes ont été enlevées, et 11 ajoutées (4 et 5 dans le kernel, respectivement). Toutefois, l’architecture générale et le principe du fonctionnement du logiciel ont été bien conservés.

Le code est structuré en 6 paquetages, dont un pour le parseur (**parser**), un pour l’interface graphique (**gui**) et un pour le coeur du logiciel (**kernel**). Egalement, trois paquetages tierce partie sont utilisés, **JavaCup**, par le parseur, et **JGraph** et **MSBChart**, par l’interface graphique. L’organisation des paquetages est représentée figure 6.4. Sur cette figure, il apparaît clairement que les objets du kernel ne communiquent avec les objets de l’interface graphique que par l’envoi de messages (événements ou exceptions). Autrement dit, le fonctionnement de la partie non graphique est totalement indépendant de la partie graphique. C’est cette organisation qui permet d’utiliser GNA aussi bien via des commandes en ligne (*i.e.* sans interface graphique) que via l’interface graphique.

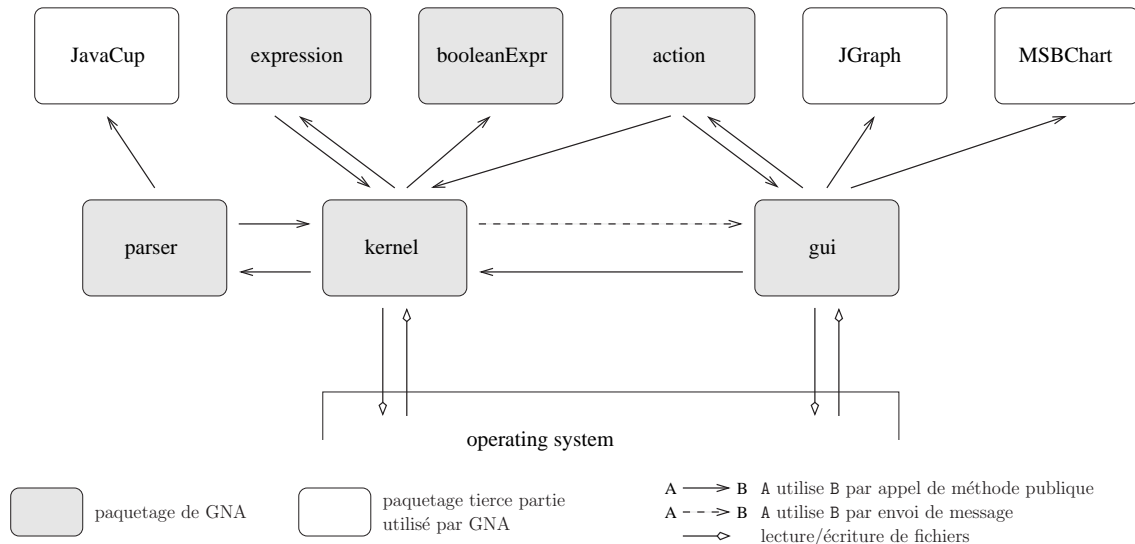


FIG. 6.4 – Organisation des paquetages constituant GNA, et leurs relations avec le système d’exploitation.

L’organisation en classes du coeur du logiciel, le kernel, est représentée figure 6.5. La classe principale est **Project**, qui est associée à un **Model**, à des **InitialConditions**, et éventuellement à des **Simulations**. Ces classes, comme toutes celles représentées figure 6.5, correspondent très naturellement aux concepts rencontrés lors de l’exposé de la méthode, dans la partie II. Une exception notable concerne les deux classes **FlowDomain** et **QState**. Effectivement, nous avons considéré dans la partie II, que les termes domaine de flot et état discret (ou qualitatif) étaient des synonymes. Ils représentent effectivement la même entité, mais qui est vue alternativement dans une représentation continue ou discrète de l’espace des phases. Du point de vue programmation, il est conceptuellement plus simple de distinguer ces deux représentations. Ainsi, un **FlowDomain** correspond à une région de

l'espace des phases, et possède donc par exemple des coordonnées, tandis qu'un `QState` correspond à un état du graphe de transitions, et possède donc par exemple des successeurs et des prédécesseurs (voir figure 6.5). Chaque `QState` est associé à un `FlowDomain` unique. Enfin, mentionnons simplement, que la représentation donnée figure 6.5 des classes du kernel est très simplifiée, son but principal étant de représenter lisiblement les classes et les relations entre classes les plus importantes pour la compréhension du fonctionnement du logiciel.

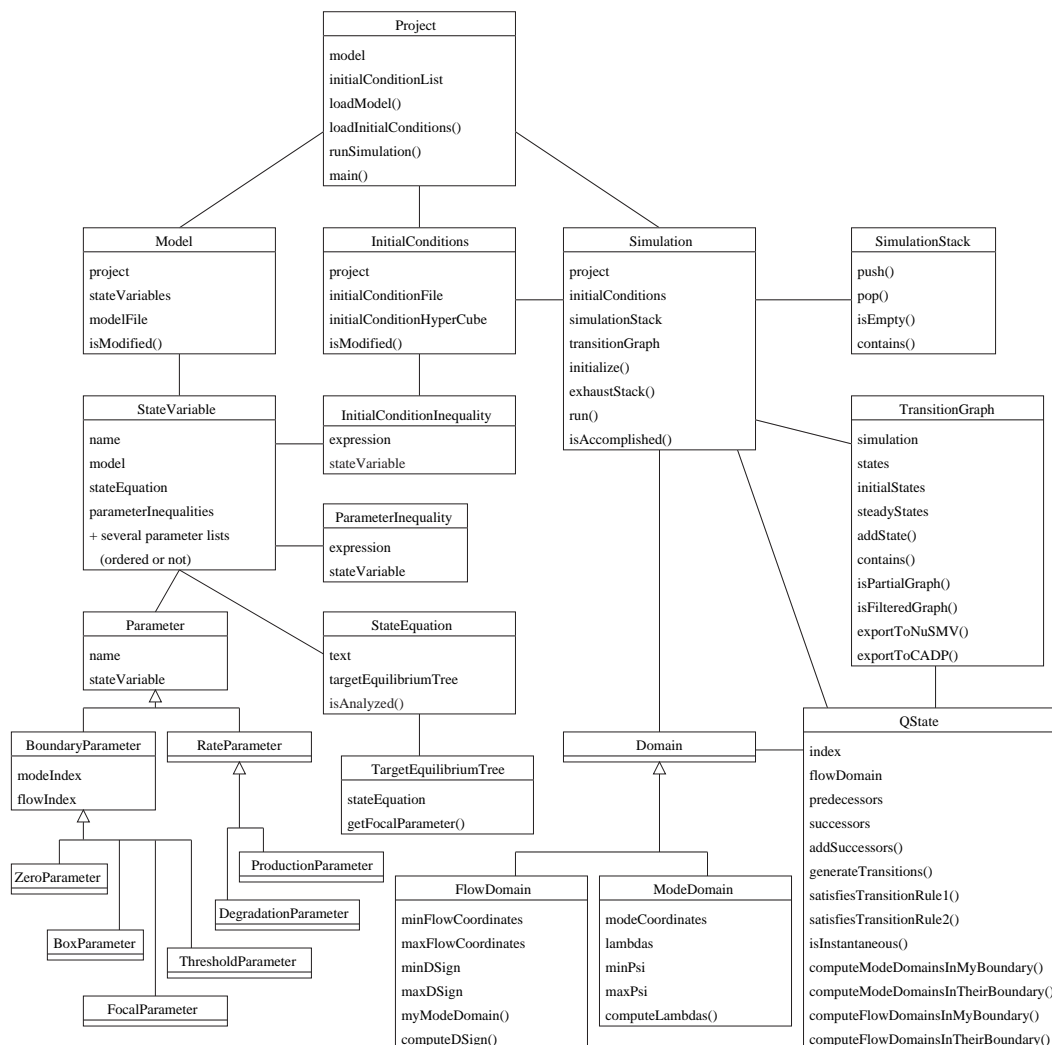


FIG. 6.5 – Diagramme de classes simplifié du kernel.

6.2.2 Aspects algorithmiques

L'objectif de cette partie n'est pas de détailler l'ensemble des algorithmes utilisés, mais seulement de donner le principe du calcul du graphe de transitions. Egalement, nous décrirons plus en détail une optimisation qui s'est révélée critique pour l'efficacité du logiciel.

Principe du calcul du graphe de transitions

Du point de vue algorithmique, le graphe de transitions est obtenu en calculant, par un parcours en profondeur, tous les états atteignables à partir des états initiaux. Les principaux objets utilisés sont `initialConditionHyperCube`, correspondant à la région initiale, hyperrectangulaire, de l'espace des phases, et `transitionGraph`, stockant les états atteints via la méthode `addState()`. Les transitions du graphe sont stockées au niveau de chaque état, dans deux listes, `successors` et `predecessors`, via la méthode `addSuccessors()`. Enfin, pour le parcours du graphe, une pile, `simulationStack`, est utilisée (voir figure 6.5).

Dans la figure 6.6, nous indiquons le pseudocode correspondant à la méthode `run()` de la classe `Simulation`. Plus précisément, `run()` appelle les méthodes `initialize()` (correspondant aux lignes 1-5) puis `exhaustStack()` (correspondant aux lignes 6-10) (voir également figure 6.5). En plus des méthodes mentionnées ci-dessus, la méthode `generateTransition()` de la classe `QState` est utilisée. Cette méthode réalise le calcul de l'ensemble des successeurs d'un état. Un algorithme pour cette méthode est présenté dans la figure 6.7.

```

run()
1: for each D in initialConditionHyperCube do
2:   D.computeDSign()
3:   QS = new QState(D)
4:   simulationStack.push(QS)
5:   transitionGraph.addState(QS)
6: while not simulationStack.isEmpty() do
7:   QS = simulationStack.pop()
8:   QS.generateTransition()
9:   for each QS' in QS.successors do
10:    if not(transitionGraph.contains(QS') or simulationStack.contains(QS'))
11:      then
12:        simulationStack.push(QS')

```

FIG. 6.6 – Pseudocode donnant l'algorithme de calcul du graphe de transitions (méthode `run()` de la classe `Simulation`).

Deux remarques peuvent être faites au sujet de ces méthodes.

Premièrement, la méthode `computeDSign()` implémente les propositions 4 et 5, et les méthodes `satisfiesTransitionRule1()` et `satisfiesTransitionRule2()` implémentent les propositions 7 et 8, respectivement (voir section 4.4).

Deuxièmement, il est important de préciser que le calcul des voisins d'un domaine de flot dans l'espace des phases (*i.e.* ligne 1 de `generateTransition()`, figure 6.7) est une tâche non triviale. Elle nécessite de calculer la partition d'un voisinage de `D` en domaines de flot dans un espace des phases pouvant être de dimension élevée. Nous allons voir au paragraphe suivant qu'une implémentation efficace de ce calcul est critique pour le passage à l'échelle de l'outil. Ce problème s'est posé de façon beaucoup moins aiguë lors de l'implémentation de la version précédente de GNA, car la partition utilisée, en domaine de mode, est bien plus simple.

```

generateTransition()
1: for each D' neighbor of D do
2:   D'.computeDSign()
3:   QS' = new QState(D')
4:   if satisfiesTransitionRule1(QS') then
5:     /* dim+ transition from QS to QS' */
6:     if not transitionGraph.contains(QS') then
7:       transitionGraph.addState(QS')
8:       addSuccessors(QS')
9:   if satisfiesTransitionRule2(QS') then
10:    /* dim- transition from QS to QS' */
11:    if not transitionGraph.contains(QS') then
12:      transitionGraph.addState(QS')
13:    addSuccessors(QS')

```

FIG. 6.7 – Pseudocode donnant un algorithme pour la méthode `generateTransition()` de la classe `QState`. D est le `FlowDomain` associé à `QS`.

Calcul efficace des successeurs d'un état

Comme nous l'avons vu dans le paragraphe précédent, pour calculer les successeurs d'un état, *i.e.* d'un domaine de flot, il est nécessaire de calculer dans un premier temps les domaines voisins du domaine considéré. Cela revient à calculer la partition d'un voisinage du domaine. L'ensemble des domaines voisins sont alors considérés comme des successeurs potentiels. L'existence d'une transition entre le domaine considéré et un successeur potentiel est alors déterminée en appliquant les règles de transitions. Ici, nous nous focalisons sur le problème du calcul de l'ensemble des successeurs potentiels. Deux méthodes, appelées dans la suite « naïve » et « optimisée », ont été successivement implémentées dans GNA. Leur principe est illustré à partir d'un exemple. Nous considérons à nouveau l'exemple du réseau d'inhibition réciproque (figure 3.1), et pour ce réseau, nous souhaitons calculer les voisins de l'état $D^{12.1}$, représenté figure 6.8(a).

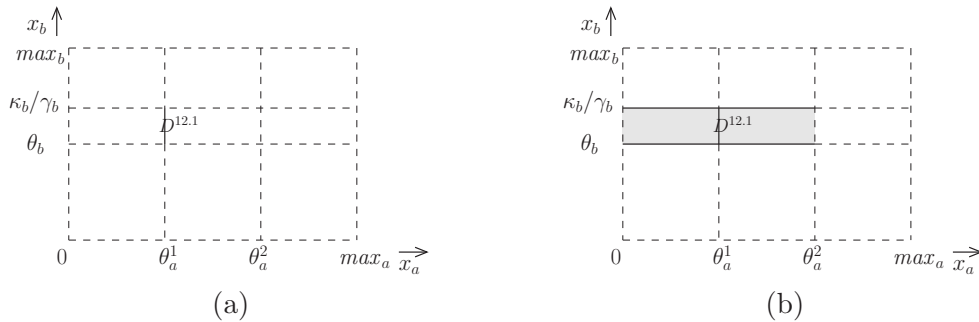


FIG. 6.8 – (a) Problème du calcul des voisins de $D^{12.1}$. (b) Représentation d'une région hyperrectangulaire « entourant » $D^{12.1}$ pouvant servir à définir un ensemble de successeurs potentiels de $D^{12.1}$.

Une implémentation naïve part du constat que nécessairement, les successeurs poten-

tiels d'un domaine D sont dans, ou intersectent avec, une région hyperrectangulaire Z qui « entoure » le domaine, c'est-à-dire telle que $D_i \subsetneq Z_i$. Ainsi, pour $D^{12.1} = \{\theta_a^1\} \times]\theta_b, \kappa_b/\gamma_b[$, une région hyperrectangulaire Z possible est $]0, \theta_a^2[\times]\theta_b, \kappa_b/\gamma_b[$. Cette région est représentée figure 6.8(b). On trouve alors comme ensemble de successeurs potentiels les 7 domaines de flot représentés figure 6.9. L'application des règles de transition entre $D^{12.1}$ et ces domaines montre qu'un seul domaine est effectivement un successeur de $D^{12.1}$, à savoir $D^{11.2}$.

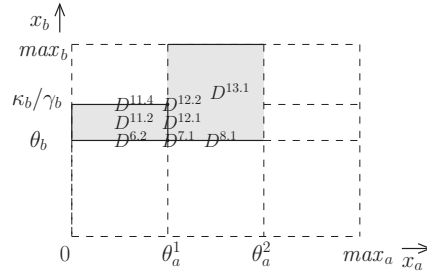


FIG. 6.9 – Ensemble de successeurs potentiels de $D^{12.1}$ obtenu par la méthode naïve.

Nous pouvons observer que parmi les domaines considérés comme successeurs potentiels, certains peuvent être exclus d'office : $D^{6.2}$, $D^{8.1}$ et $D^{11.4}$. Effectivement, aucun de ces domaines n'ont $D^{12.1}$ dans leur frontière ou ne sont dans la frontière de $D^{12.1}$. Pourtant, ce sont respectivement des conditions nécessaires à l'existence d'une transition dim^+ ($D \subseteq \partial D'$ dans la proposition 7) ou dim^- ($D' \subseteq \partial D$ dans la proposition 8).

Ce constat simple suggère d'employer une autre méthode pour calculer un ensemble, plus restreint, de successeurs potentiels d'un état : considérer deux régions séparément, l'une, Z^1 , contenant, ou intersectant avec, les domaines D' tels que $D \subseteq \partial D'$, et l'autre, Z^2 , contenant, ou intersectant avec, les domaines D' tels que $D' \subseteq \partial D$. Autrement dit, Z^1 satisfait $D_i \subsetneq Z_i^1$, si D_i est un singleton, et $D_i = Z_i^1$, sinon, et Z^2 satisfait $D_i = Z_i^2$, si D_i est un singleton, et $D_i \subsetneq Z_i^2$, sinon.

Pour $D^{12.1}$, les deux régions $]0, \theta_a^2[\times]\theta_b, \kappa_b/\gamma_b[$, représentée figure 6.10(a), et $\{\theta_a^1\} \times]\theta_b, \kappa_b/\gamma_b[$, représentée figure 6.10(b), satisfont les critères donnés pour Z^1 et Z^2 , respectivement. On peut noter que l'union de ces deux régions est strictement plus petite que la région Z représentée figure 6.9. L'ensemble des successeurs potentiels obtenus est alors $D^{7.1}$, $D^{11.2}$, $D^{12.2}$ et $D^{13.1}$, comme représenté figure 6.11.

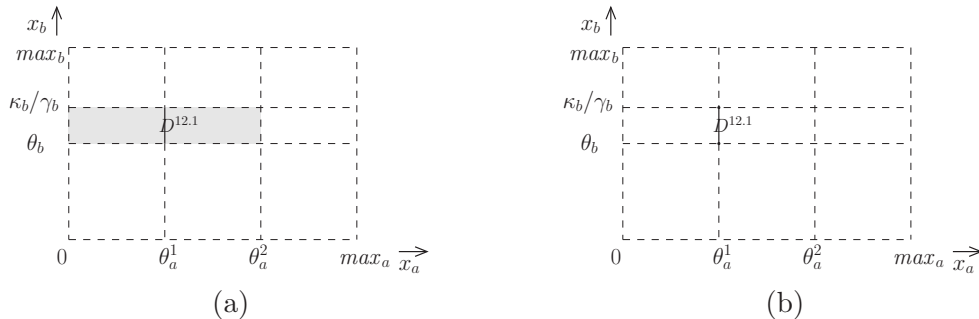
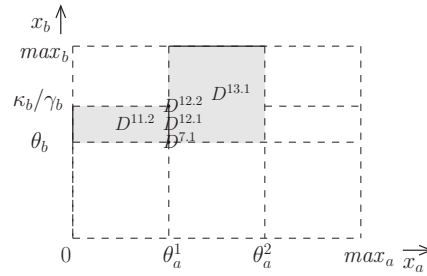


FIG. 6.10 – Représentation de deux régions hyperrectangulaires incluant $D^{12.1}$ et pouvant servir à définir deux ensembles de successeurs potentiels D' de $D^{12.1}$ tels que (a) $D^{12.1} \subseteq \partial D'$ et (b) $D' \subseteq \partial D^{12.1}$.

FIG. 6.11 – Ensemble de successeurs potentiels de $D^{12.1}$ obtenu par la méthode optimisée.

modèle	nombre de variables	conditions initiales	nombre d'états atteignables	temps de calcul méthode naïve	temps de calcul méthode optimisée
xaxb	2 (2)	Ω	27	0.19 s	0.01 s
tutorial	3 (3)	Ω	229	0.77 s	0.04 s
delta-notch03	4 (4)	Ω	533	11.4 s	0.19 s
coli43	6 (7)	D^{init}	201	81.3 s	0.19 s
bacillus35-kinA-hpr	7 (11)	D^{init}	3	4173.2 s	0.06 s
bacillus35	9 (11)	D^{init}	1143	?	20.8 s

TAB. 6.1 – Comparaison de deux implémentations de GNA. Le nombre de variables d'état (et total) est indiqué pour chaque modèle. Les conditions initiales Ω ou D^{init} indiquent respectivement qu'une analyse de l'espace des phases ou qu'une analyse d'atteignabilité à partir d'un unique état initial est réalisée.

L'implémentation de la seconde approche repose sur l'utilisation de quatre itérateurs de la classe `QState`, appelés `modeDomainsInMyBoundary()`, `flowDomainsInMyBoundary()`, `modeDomainsInTheirBoundary()` et `flowDomainsInTheirBoundary()` (voir figure 6.5). Ces méthodes sont spécialisées et complexes. Les deux premières (dernières) méthodes permettent d'énumérer les domaines de mode, puis les domaines de flot, qui sont dans, ou qui intersectent avec, une région de type Z^2 (Z^1). Seuls les domaines D' satisfaisant soit $D \subseteq \partial D'$, soit $D' \subseteq \partial D$, sont générés par cette approche.

Pour notre exemple, le nombre de successeurs potentiels considérés est relativement similaire que l'on utilise la méthode naïve ou optimisée (4 ou 7 domaines). Toutefois, la seconde méthode devient considérablement plus efficace que la première lorsque la dimension de l'espace des phases, *i.e.* le nombre de variables, croît. Les deux méthodes ont été implémentées dans deux versions de GNA. Leurs efficacités sont comparées dans la table 6.1. Les modèles utilisés sont disponibles dans la distribution, ou sur demande. Les résultats sont obtenus sur un PC (800MHz, 256Mo). Ces données montrent très clairement que la seconde implémentation supporte largement mieux la mise à l'échelle. Le calcul efficace des successeurs potentiels est critique pour l'efficacité du logiciel.

Dans cette partie, nous avons présenté l'outil informatique *Genetic Network Analyzer*, version 6.0 (GNA 6.0), qui implémente la méthode présentée dans la partie II.

La fonction principale de GNA est de calculer le système de transitions LPM qualitatif, aussi appelé graphe de transitions, associé à un modèle LPM qualitatif. L'analyse peut être restreinte à la partie du graphe atteignable à partir d'un ensemble d'états initiaux. D'importantes optimisations sont permises par le fait que GNA soit dédié à ce type de cal-

cul. GNA offre également diverses possibilités pour l'analyse des résultats de simulation. Premièrement, un ensemble d'opérations peuvent être réalisées sur le graphe de transitions via une interface graphique. Deuxièmement, des possibilités d'export vers des model checkers, permettent d'analyser de façon efficace les propriétés dynamiques du système.

Dans la partie suivante, nous présentons deux applications à l'analyse de phénomènes bactériens complexes. Ces applications illustrent les capacités de l'outil à analyser des modèles grands et complexes, et plus généralement, illustrent l'intérêt biologique de la méthode présentée dans la partie II et de son implémentation dans l'outil GNA.

Quatrième partie

Applications

Analyse de l'initiation de la sporulation chez *B. subtilis*

Dans ce chapitre et dans le chapitre suivant, nous utilisons la méthode et l'outil présentés dans les parties précédentes (parties II et III) pour tester la validité de modèles de réseaux de régulation génique en comparant les prédictions obtenues avec des données expérimentales disponibles dans la littérature. Ces deux chapitres ont un double objectif. Premièrement, démontrer l'applicabilité de l'approche proposée à l'analyse de systèmes d'intérêt biologique, et deuxièmement, contribuer à une meilleure compréhension des phénomènes étudiés en testant la validité de modèles grands et complexes de ces phénomènes.

Dans ce chapitre, nous nous intéressons plus spécifiquement à la sporulation chez *Bacillus subtilis*, un phénomène bactérien d'adaptation à des changements environnementaux. Cette réponse implique d'importantes modifications morphologiques et physiologiques, et est contrôlée principalement au niveau génétique. Proposer une réponse adéquate, ni trop précoce, ni trop tardive, est souvent critique pour la survie de la lignée cellulaire. Pour tenter de mieux comprendre ce phénomène, un modèle LPM qualitatif du réseau de régulation génique contrôlant l'initiation de la sporulation a été précédemment construit par de Jong *et al.* [67] et analysé avec la méthode proposée dans [70]. Dans ce chapitre, nous utilisons la méthode présentée dans la partie II pour tester la validité de ce modèle.

Dans la première section, nous présentons plus en détail le phénomène de la sporulation chez *B. subtilis*, ainsi que le modèle proposé dans [67]. Puis, nous utilisons la méthode d'analyse qualitative présentée dans la partie II pour simuler la réponse de différentes souches de *B. subtilis* (*i.e.* la souche sauvage et des mutants) placées dans des conditions environnementales défavorables. Une comparaison des résultats obtenus avec les deux méthodes, celle proposée par de Jong *et al.* [70] et celle que nous proposons (partie II), fait apparaître que premièrement, l'utilisation d'une partition plus fine permet effectivement d'obtenir une description significativement plus précise du comportement du système et que deuxièmement, l'augmentation du nombre des états atteignables reste modérée. Les prédictions obtenues sont alors utilisées dans la section 7.3 pour tester la validité du modèle, en les comparant avec des données de la littérature. Un certain nombre de propriétés sur la dynamique du système sont formalisées en logique temporelle et leur cohérence avec les prédictions est testée par model checking. Nous montrons que les prédictions obtenues sont globalement en accord avec les données expérimentales. Toutefois, quelques incohérences entre prédictions et observations sont également identifiées, qui nous conduisent soit à suggérer des extensions du modèle, soit à proposer la réalisation d'expériences complémentaires.

7.1 Modélisation de l'initiation de la sporulation

7.1.1 Sporulation chez *B. subtilis*

Comme toute bactérie, *B. subtilis* est en compétition constante avec les individus de son espèce, ou d'une autre espèce, pour l'accès à des ressources limitées. Tant que des

nutriments sont disponibles, les bactéries croissent et se divisent aussi vite que possible. Toutefois, lorsque les conditions environnementales se détériorent, les cellules cessent leur croissance et entrent en phase stationnaire. Lors de l'entrée en phase stationnaire, des programmes d'acquisition de résistances, destinés à permettre la survie dans un environnement de plus en plus hostile, sont initiés [107, 114, 203]. La réponse ultime des cellules est la sporulation [83, 107, 123, 205, 206], c'est-à-dire la formation de spores qui sont remarquablement résistantes aux attaques chimiques, à la chaleur, à la dessiccation et aux radiations. Les spores peuvent rester dormantes pendant plusieurs années puis germer lorsque les conditions environnementales redeviennent favorables. La cellule reprend alors sa croissance végétative. Dans une culture bactérienne appauvrie en nutriments, la plupart des cellules sporulent (phénotype spo^+), tandis qu'une minorité continue la croissance végétative (phénotype spo^-).

La décision d'abandonner la croissance végétative et d'initier la sporulation implique une modification radicale de l'expression des gènes dans la cellule. Cette modification de l'expression génique est contrôlée par un réseau de régulation génique complexe, impliquant plus de 125 gènes [206]. La facilité avec laquelle il est possible de manipuler génétiquement *B. subtilis* a permis d'identifier et de caractériser un grand nombre de ces gènes et des interactions géniques composant ce réseau. Le processus de prise de décision entre l'entrée en sporulation et la poursuite de la croissance végétative est ainsi devenu un exemple paradigmatique permettant de mieux comprendre les processus du développement des organismes.

7.1.2 Réseau de régulation génique contrôlant l'initiation de la sporulation

Une étape essentielle dans l'engagement de la cellule dans la voie de la sporulation est la phosphorylation d'un facteur de transcription, Spo0A, en réponse à différents signaux environnementaux, métaboliques ou du cycle cellulaire. Au-dessus d'un certain seuil de concentration, la protéine phosphorylée, Spo0A~P, déclenche une cascade d'activation de facteurs sigma qui contrôlent la transcription des gènes responsables des modifications morphologiques et physiologiques ayant lieu lors de la sporulation. Ainsi, la phosphorylation de Spo0A permet la synthèse de σ^H , impliqué dans la formation du septum. Également, Spo0A~P active la synthèse des facteurs σ^E et σ^F , qui jouent un rôle clé dans les développements différents de la cellule-mère et de la préspore, respectivement. Nous nous intéressons ici aux étapes initiales de la sporulation et nous considérerons que la cellule est engagée dans la voie de la sporulation dès que la concentration de σ^F dépasse un certain seuil.

Dans la figure 7.2 sont représentés des composants, gènes et protéines, impliqués dans l'initiation de la sporulation, ainsi que leurs interactions. Nous utilisons les conventions graphiques proposées par Kohn [137] (figure 7.1). Le réseau de la sporulation est centré autour du phosphorelais, responsable de la phosphorylation de Spo0A [107, 123]. Plus précisément, l'état de phosphorylation de Spo0A dépend de l'activité antagoniste de diverses kinases et phosphatases, agissant sur Spo0A, soit directement, soit via la phosphorylation de Spo0B et Spo0F. L'activité de ces kinases et phosphatases dépend elle-même, selon des mécanismes encore mal compris, de nombreux facteurs tels que la disponibilité de nutriments, la densité de la population, l'activité métabolique et la progression dans le cycle cellulaire. Le phosphorelais est donc le siège de l'intégration de nombreux signaux. Ici, nous simplifions le phosphorelais en ne considérant qu'une seule kinase, KinA, et qu'une seule

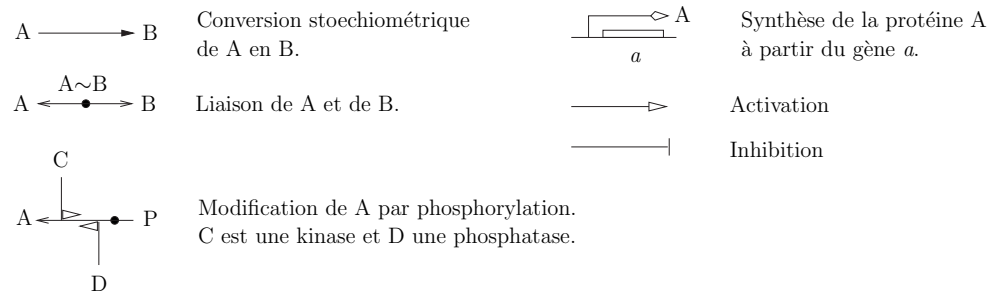


FIG. 7.1 – Conventions graphiques utilisées pour représenter les réseaux de régulation génique. [137].

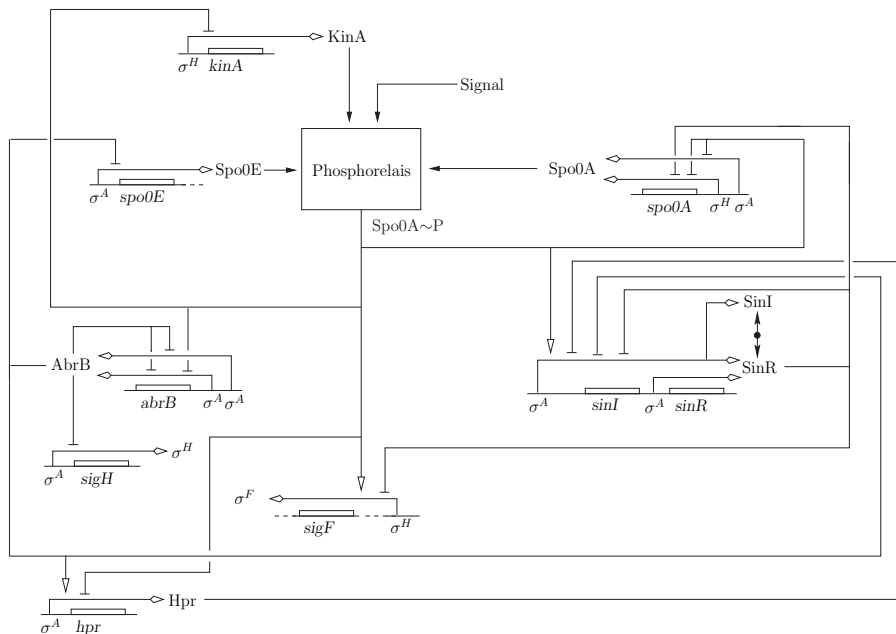


FIG. 7.2 – Gènes, protéines et interactions clés, impliqués dans l’initiation de la sporulation chez *B. subtilis*. Les conventions graphiques utilisées sont données dans la figure 7.1. Le contenu de la boîte appelée Phosphorelais est détaillé dans la figure 7.3. Afin d’améliorer la lisibilité de la figure, le contrôle de la transcription par les facteurs σ^A et σ^H est représenté implicitement en étiquetant les promoteurs avec le facteur sigma approprié. [67].

phosphatase, Spo0E, et en ignorant les étapes intermédiaires du phosphorelais, c'est-à-dire la phosphorylation de Spo0F et de Spo0B (figure 7.3). Ces simplifications ne modifient pas la fonction essentielle du phosphorelais : moduler le niveau de phosphorylation de Spo0A en fonction des actions antagonistes des kinases et phosphatases [178]. Egalement, nous considérons un unique signal environnemental, appelé simplement Signal, qui active KinA en condition de carences nutritionnelles.

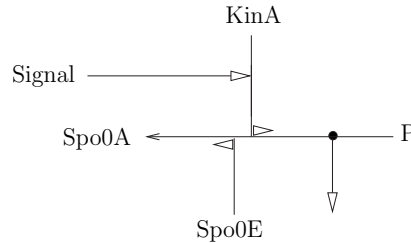


FIG. 7.3 – Description simplifiée du phosphorelais impliqué dans l'initiation de la sporulation chez *B. subtilis*. Seule une kinase (KinA), une phosphatase (Spo0E) et un signal environnemental (Signal) est pris en compte. Les étapes de la phosphorylation de Spo0F et de Spo0B sont ignorées. [67].

Les composants du phosphorelais sont eux-même régulés au niveau transcriptionnel par Spo0A~P, et par d'autres protéines régulées directement ou indirectement par Spo0A~P. Par exemple, AbrB, contrôlé par Spo0A, inhibe l'expression de *spo0E* et, via le contrôle des gènes de l'opéron *sin*, *sinI* et *sinR*, régule l'expression de *kinA* et de *spo0A*. AbrB inhibe également l'expression du facteur sigma σ^H , impliqué dans l'expression des gènes *sinI*, *sinR* et *hpr*. L'expression de *sigF*, codant le facteur sigma σ^F que l'on considère comme le marqueur de l'engagement de la cellule dans la voie de la sporulation, est contrôlée à la fois par Spo0A et par AbrB. En plus de ces interactions, plusieurs auto-inhibitions ont lieu (pour *spo0A*, *abrB* et l'opéron *sin*). Une description plus complète des interactions géniques du réseau est donnée dans [67].

Comment la décision d'entrer en sporulation ou de poursuivre la croissance végétative émerge de ces interactions ? La description des différentes interactions proposée ci-dessus, bien que partielle et très simplifiée, donne déjà une idée de la difficulté de prédire le comportement global du système par des raisonnements intuitifs. De plus, bien que ce phénomène ait fait l'objet d'études intensives pendant plusieurs décennies, très peu de données numériques sont disponibles pour les valeurs des paramètres cinétiques et des concentrations moléculaires, rendant difficile l'utilisation de méthodes d'analyse numérique. Aussi, afin de pouvoir comprendre comment la prise de décision émerge des interactions entre composants du réseau, un modèle LPM qualitatif a été construit.

7.1.3 Modèle LPM du réseau contrôlant l'initiation de la sporulation

A partir de l'abondante littérature et de bases de données telles que SubtiList [169], un modèle LPM qualitatif a été construit, incluant les gènes codant σ^A , σ^F , σ^H , Spo0A, KinA, Spo0E, AbrB, SinI, SinR et Hpr, et un signal de sporulation, Signal [67].

Ce modèle consiste en 11 équations différentielles LPM, et autant d'ensembles de contraintes d'inégalité sur les paramètres, représentant en tout 57 inégalités. Le modèle est présenté figure 7.4. Les fonctions de régulation et les contraintes d'inégalité ont été spécifiées sur la base des informations génétiques et moléculaires disponibles dans la littérature.

$\dot{u}_a = 0$ $0 < \theta_a < \max_a$
$\dot{x}_h = \kappa_h^0 s^+(x_a, \theta_a) + \kappa_h^1 s^+(x_a, \theta_a) s^-(x_{ab}, \theta_{ab}^1) - \gamma_h x_h$ $0 < \kappa_h^0/\gamma_h < \theta_h < (\kappa_h^0 + \kappa_h^1)/\gamma_h < \max_h$
$\dot{x}_f = \kappa_f s^+(x_h, \theta_h) s^+(u_s, \theta_s) s^+(x_{sa}, \theta_{sa}^3) s^+(x_{ka}, \theta_{ka}^3) s^-(x_{se}, \theta_{se}^2) (1 - s^+(x_{sr}, \theta_{sr}^1) s^-(x_{si}, \theta_{si}^2)) - \gamma_f x_f$ $0 < \theta_f < \kappa_f/\gamma_f < \max_f$
$\dot{x}_{sa} = \kappa_{sa}^1 s^+(x_a, \theta_a) (1 - s^+(u_s, \theta_s) s^+(x_{sa}, \theta_{sa}^3) s^+(x_{ka}, \theta_{ka}^3) s^-(x_{se}, \theta_{se}^2))$ $+ \kappa_{sa}^2 s^+(x_h, \theta_h) (1 - s^+(u_s, \theta_s) s^+(x_{sa}, \theta_{sa}^4) s^+(x_{ka}, \theta_{ka}^4) s^-(x_{se}, \theta_{se}^1))$ $\cdot (1 - s^+(x_{sr}, \theta_{sr}^1) s^-(x_{si}, \theta_{si}^2)) - \gamma_{sa} x_{sa}$ $0 < \theta_{sa}^1 < \theta_{sa}^2 < \kappa_{sa}^1/\gamma_{sa} < \theta_{sa}^3 < \theta_{sa}^4 < \kappa_{sa}^2/\gamma_{sa} < (\kappa_{sa}^1 + \kappa_{sa}^2)/\gamma_{sa} < \max_{sa}$
$\dot{x}_{ka} = \kappa_{ka}^0 + \kappa_{ka}^1 s^+(x_h, \theta_h) (1 - s^+(u_s, \theta_s) s^+(x_{sa}, \theta_{sa}^4) s^+(x_{ka}, \theta_{ka}^4) s^-(x_{se}, \theta_{se}^1)) - \gamma_{ka} x_{ka}$ $0 < \theta_{ka}^1 < \theta_{ka}^2 < \kappa_{ka}^0/\gamma_{ka} < \theta_{ka}^3 < \theta_{ka}^4 < \kappa_{ka}^1/\gamma_{ka} < (\kappa_{ka}^0 + \kappa_{ka}^1)/\gamma_{ka} < \max_{ka}$
$\dot{x}_{se} = \kappa_{se}^0 s^+(x_a, \theta_a) + \kappa_{se}^1 s^+(x_a, \theta_a) s^-(x_{ab}, \theta_{ab}^1) - \gamma_{se} x_{se}$ $0 < \theta_{se}^1 < \kappa_{se}^0/\gamma_{se} < \theta_{se}^2 < \theta_{se}^3 < (\kappa_{se}^0 + \kappa_{se}^1)/\gamma_{se} < \theta_{se}^4 < \max_{se}$
$\dot{x}_{ab} = \kappa_{ab}^1 s^+(x_a, \theta_a) s^-(x_{ab}, \theta_{ab}^2) + \kappa_{ab}^2 s^+(x_a, \theta_a) s^-(x_{ab}, \theta_{ab}^2)$ $\cdot (1 - s^+(u_s, \theta_s) s^+(x_{sa}, \theta_{sa}^1) s^+(x_{ka}, \theta_{ka}^1) s^-(x_{se}, \theta_{se}^4)) - \gamma_{ab} x_{ab}$ $0 < \kappa_{ab}^1/\gamma_{ab} < \theta_{ab}^1 < \theta_{ab}^2 < (\kappa_{ab}^1 + \kappa_{ab}^2)/\gamma_{ab} < \max_{ab}$
$\dot{x}_{sr} = \kappa_{sr}^0 s^+(x_a, \theta_a) + \kappa_{sr}^1 s^+(x_a, \theta_a) s^-(x_{ab}, \theta_{ab}^1) s^-(x_{hr}, \theta_{hr}) (1 - s^+(x_{sr}, \theta_{sr}^2) s^-(x_{si}, \theta_{si}^1))$ $\cdot s^+(u_s, \theta_s) s^+(x_{sa}, \theta_{sa}^2) s^+(x_{ka}, \theta_{ka}^2) s^-(x_{se}, \theta_{se}^3) - \gamma_{sr} x_{sr}$ $0 < \theta_{sr}^1 < \theta_{sr}^2 < \kappa_{sr}^0/\gamma_{sr} < (\kappa_{sr}^0 + \kappa_{sr}^1)/\gamma_{sr} < \max_{sr}$
$\dot{x}_{si} = \kappa_{si}^0 s^+(x_a, \theta_a) + \kappa_{si}^1 s^+(x_a, \theta_a) s^-(x_{ab}, \theta_{ab}^1) s^-(x_{hr}, \theta_{hr}) (1 - s^+(x_{sr}, \theta_{sr}^2) s^-(x_{si}, \theta_{si}^1))$ $\cdot s^+(u_s, \theta_s) s^+(x_{sa}, \theta_{sa}^2) s^+(x_{ka}, \theta_{ka}^2) s^-(x_{se}, \theta_{se}^3) - \gamma_{si} x_{si}$ $0 < \theta_{si}^1 < \kappa_{si}^0/\gamma_{si} < \theta_{si}^2 < (\kappa_{si}^0 + \kappa_{si}^1)/\gamma_{si} < \max_{si}$
$\dot{x}_{hr} = \kappa_{hr} s^+(x_a, \theta_a) s^+(x_{ab}, \theta_{ab}^1) (1 - s^+(u_s, \theta_s) s^+(x_{sa}, \theta_{sa}^2) s^+(x_{ka}, \theta_{ka}^2) s^-(x_{se}, \theta_{se}^3)) - \gamma_{hr} x_{hr}$ $0 < \theta_{hr} < \kappa_{hr}/\gamma_{hr} < \max_{hr}$
$\dot{u}_s = 0$ $0 < \theta_s < \max_s$

FIG. 7.4 – Equations différentielles LPM et contraintes d’inégalité sur les paramètres pour le réseau de la sporulation représenté figure 7.2. Les concentrations des protéines sont décrites par u_a pour σ^A , x_h pour σ^H , x_f pour σ^F , x_{sa} pour Spo0A, x_{ka} pour KinA, x_{se} pour Spo0E, x_{ab} pour AbrB, x_{sr} pour SinR, x_{si} pour SinI, x_{hr} pour Hpr. Le signal de sporulation est représenté par u_s . [67].

L'ensemble des étapes de la construction du modèle est décrit en détail dans [67]. Précisons simplement que le signal de sporulation, activant la kinase KinA, est représenté par une variable d'entrée u_s telle que $\dot{u}_s = 0$ et ayant un seul seuil associé, θ_s . On utilise alors le fait que la fonction en escalier $s^+(u_s, \theta_s)$ vaut 1 lorsque le signal est présent ($u_s > \theta_s$), et 0 lorsque le signal est absent ($u_s < \theta_s$).

7.2 Analyse qualitative

Dans cette section, nous simulons la réponse de différentes souches de *B. subtilis*, sauvage et mutantes, placées dans des conditions environnementales défavorables. Ces résultats sont obtenus par la méthode présentée dans la partie II, qui utilise une partition fine, en domaines de flot, de l'espace des phases. Pour comparaison, les résultats obtenus par la méthode proposée par de Jong *et al.* [70], qui utilise une partition plus grossière, en domaines de mode, de l'espace des phases, sont également donnés. Une discussion des résultats obtenus par ces deux méthodes est proposée à la section 7.2.2.

7.2.1 Simulation de l'initiation de la sporulation

A l'aide du modèle, nous simulons la réponse des bactéries placées dans des conditions défavorables (manque de nutriments, densité de population élevée, ...). Lorsque le signal de sporulation est absent ($u_s < \theta_s$), mimant des conditions environnementales favorables, le système atteint un état d'équilibre (singulier) qui correspond à des conditions physiologiques de cellules en phase de croissance exponentielle :

$$\begin{aligned}
0 &\leq u_s < \theta_s, \\
\theta_a &< u_a < \max a, & x_f &= 0, \\
x_{sa} &= \kappa_{sa}^1 / \gamma_{sa}, & x_{ka} &= \kappa_{ka} / \gamma_{ka}, \\
x_{se} &= \kappa_{se}^0 / \gamma_{se}, & x_{ab} &= \theta_{ab}^2, \\
x_{sr} &= \kappa_{sr}^0 / \gamma_{sr}, & x_{si} &= \kappa_{si}^0 / \gamma_{si}, \\
x_{hr} &= \kappa_{hr} / \gamma_{hr}, & x_h &= \kappa_h^0 / \gamma_h.
\end{aligned} \tag{7.1}$$

A partir de cet état, nous simulons la présence de conditions de croissance défavorables en commutant le signal : $\theta_s < u_s \leq \max a$. L'ensemble des états atteignables à partir de l'état initial est calculé grâce à la nouvelle version de GNA, présentée dans la partie III. Un graphe de transitions contenant 1348 états est obtenu en une vingtaine de secondes (21 s) sur un PC (800MHz, 256Mo). Ce graphe est représenté dans la figure 7.5. Rappelons que formellement, ce graphe de transitions est un système de transitions LPM qualitatif, et les états du graphe sont des domaines de flot. La plupart des états sont instantanés. Etant donné que ces derniers n'ont que peu de signification biologique, ils peuvent être éliminés (voir section 5.3). On obtient ainsi un graphe de transitions, dit filtré, réduit à 249 états persistants.

Ce graphe contient un unique état d'équilibre, atteignable à partir de tous les états. Deux chemins dans le graphe de transitions filtré, correspondant à deux exécutions du système allant de l'état initial, D^2 , à l'état d'équilibre, D^{31} , sont représentés figure 7.6. Dans le premier, figure 7.6(a), on observe une expression transitoire du gène *sigF*, caractérisant un engagement de la cellule dans la voie de la sporulation (phénotype *spo*⁺) : la concentration de σ^F , x_f , dépasse le seuil θ_f . Dans le second chemin, la concentration de

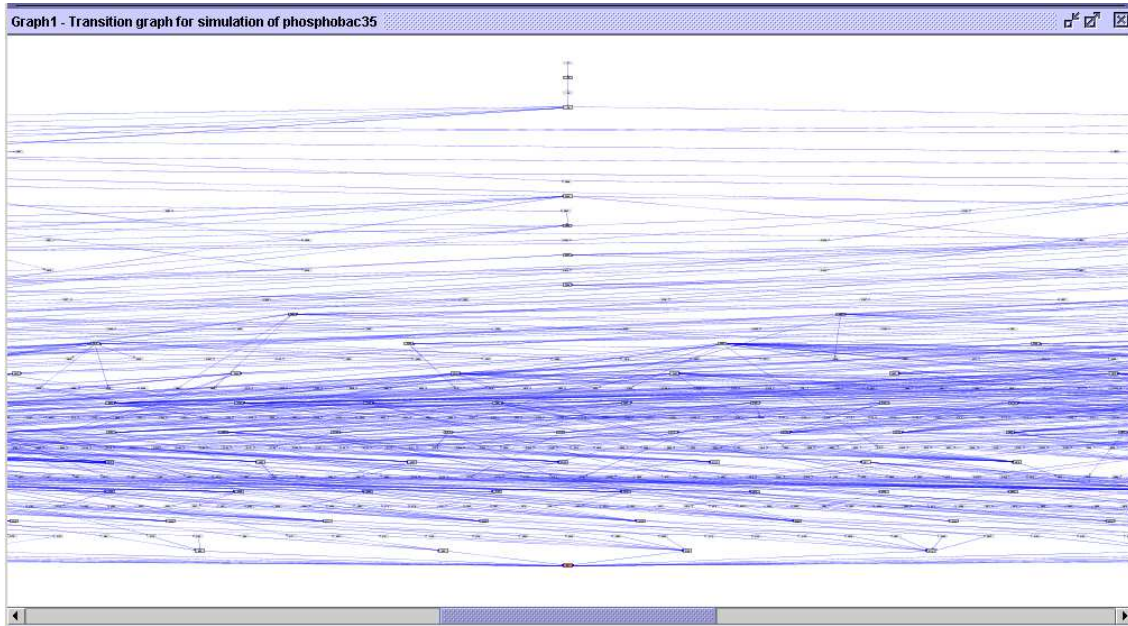


FIG. 7.5 – Graphe de transitions obtenu lors de la simulation de l’initiation de la sporulation chez *B. subtilis*.

σ^F reste toujours basse. Le phénotype correspondant est donc spo^- . Ainsi, les résultats obtenus prédisent que les deux réponses de *B. subtilis*, l’entrée en sporulation et la continuation de la croissance végétative, sont possibles en conditions défavorables. Comme nous le verrons dans la section suivante, ceci est tout à fait cohérent avec ce qui est observé expérimentalement [196].

Une grande partie des connaissances accumulées sur l’architecture et le fonctionnement du réseau représenté figure 7.2 provient de l’analyse de mutants. A partir du modèle de la souche sauvage, il est facile de créer un modèle pour les mutants, en modifiant les équations d’état ou les contraintes d’inégalité sur les paramètres. Par exemple, une délétion du gène *sigH* se traduit dans le modèle par $\dot{x}_h = -\gamma_h x_h$ (seule la dégradation de la protéine demeure). Les conditions initiales doivent en général être adaptées en conséquence.

Une analyse similaire à celle détaillée plus haut pour la souche sauvage peut être faite pour un grand nombre de mutants. Les résultats de l’analyse de différents mutants sont synthétisés dans la table 7.1.

7.2.2 Apport de l’utilisation de la partition en domaines de flot

Comme nous l’avons dit plus haut, la méthode présentée dans la partie II et utilisée ici est un raffinement de la méthode proposée par de Jong *et al.* [70]. Tandis que l’abstraction discrète utilisée par la première repose sur une partition de l’espace des phases en domaines de flot, celle utilisée dans la seconde repose sur une partition en domaines de mode. On peut alors se demander, premièrement, si l’utilisation de la méthode plus fine améliore effectivement la précision des prédictions, de façon à permettre une comparaison plus discriminante avec les données expérimentalement disponibles, et deuxièmement, si elle n’entraîne pas une augmentation exponentielle du nombre d’états (*i.e.* de domaines) atteignables, préjudiciable à la mise à l’échelle de la méthode.

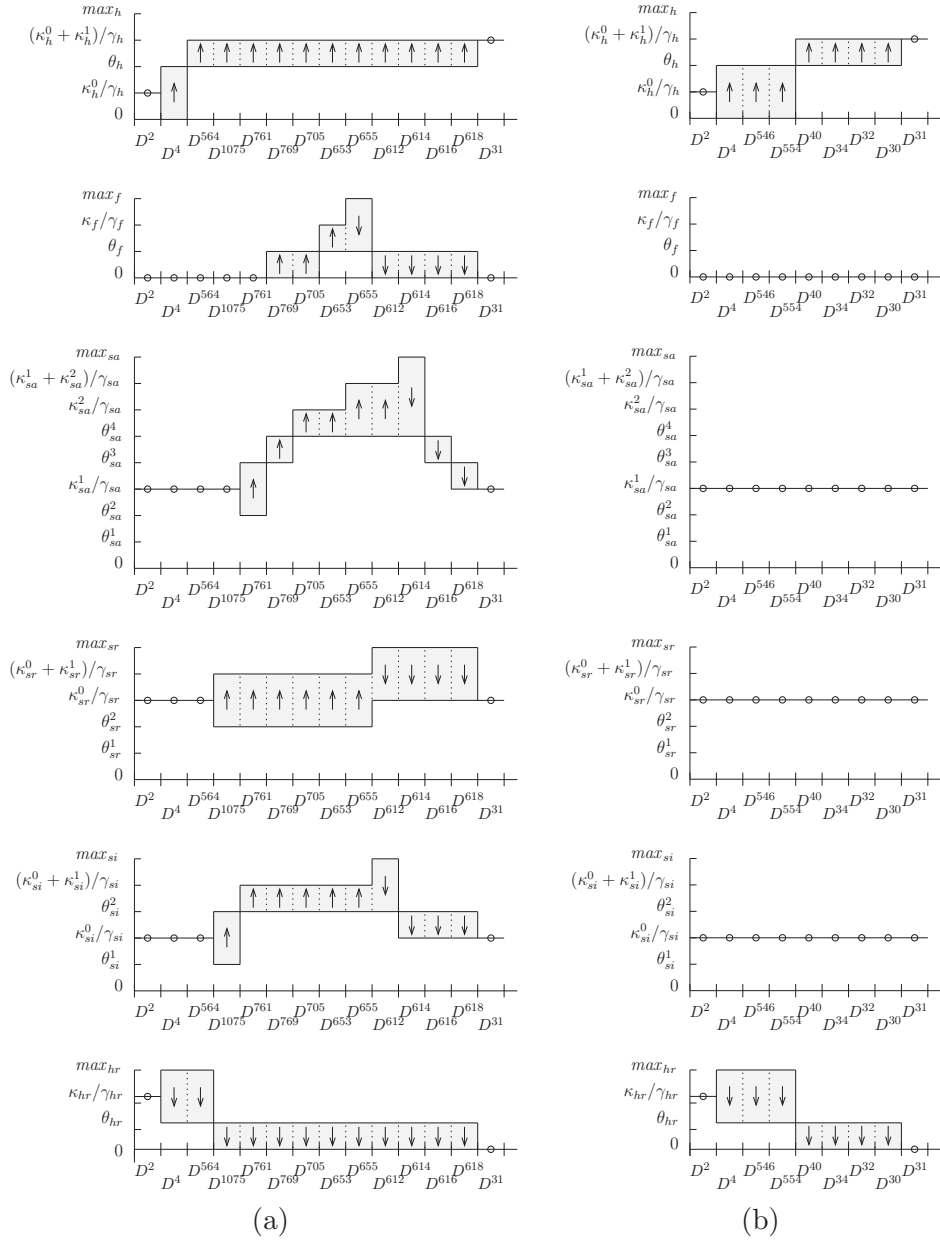


FIG. 7.6 – Evolution temporelle de la concentration de certaines protéines du réseau le long de deux chemins du graphe de transitions, lors de la simulation de l'initiation de la sporulation chez *B. subtilis* : (a) chemin correspondant à un phénotype spo^+ ; (b) chemin correspondant à un phénotype spo^- . Les parties grisées indiquent les plages de concentration des protéines et les flèches, les sens de variation des concentrations (\uparrow : augmente, \downarrow : diminue, et \circ : constant).

Modification génétique	Nombre d'états	Nombre d'états persistants	Phénotype prédit	Temps de calcul
sauvage	1348 (465)	249 (82)	spo^+ , spo^-	21 s (8 s)
mutant $spo0A$	1 (2)	1 (2)	spo^-	<1 s (<1 s)
mutant $sigH$	19 (17)	8 (7)	spo^-	<1 s (<1 s)
mutant $sinI$	139 (137)	32 (31)	spo^-	<1 s (<1 s)
mutant $kinA$	1 (2)	1 (2)	spo^-	<1 s (<1 s)
mutant $sinR$	2234 (1257)	289 (142)	spo^+ , spo^-	14 s (12 s)
mutant $abrB$	1345 (463)	242 (81)	spo^+ , spo^-	3 s (4 s)
mutant $spo0A\ abrB$	136 (135)	31 (30)	spo^-	<1 s (<1 s)
surexpression d' hpr	49 (47)	17 (16)	spo^-	<1 s (<1 s)

TAB. 7.1 – Résultats de la simulation de la réponse de différentes souches de *B. subtilis*, sauvage et mutantes, placées dans des conditions environnementales défavorables. Les données entre parenthèses correspondent aux résultats obtenus avec la méthode proposée par de Jong *et al.* [70].

La description du comportement du système représenté figure 7.6(a), que l'on obtient en utilisant la méthode proposée par de Jong *et al.* [70] est donnée pour la protéine SinR dans la figure 7.7(a). Sur la base des informations prédites, il n'est par exemple pas possible d'exclure le fait que la concentration de SinR puisse continuellement décroître le long du chemin ($M^2, M^4, \dots, M^{618}, M^{31}$). Or, comme le démontre l'approche fine (figure 7.7(b)), ceci n'est pas possible. Ainsi, des incohérences entre les prédictions et les données expérimentales ne sont pas détectées par une méthode trop conservatrice. Une autre illustration de l'intérêt d'utiliser une analyse plus fine est donnée dans Batt *et al.* [25], pour un autre exemple, concernant la réponse au stress nutritionnel chez *E. coli*.

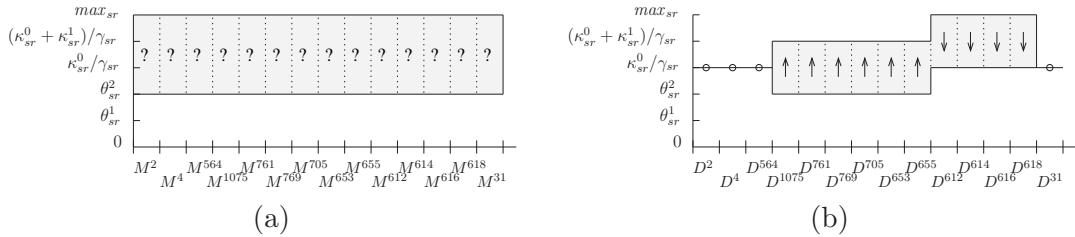


FIG. 7.7 – Comparaison des prédictions obtenues par deux méthodes d'analyse qualitative de modèles LPM. Evolution temporelle prédite pour la concentration de la protéine SinR le long du chemin présenté à la figure 7.6(a), (a) obtenue avec la partition en domaines de mode proposée par de Jong *et al.* [70], ou (b) obtenue avec la partition en domaines de flot que nous proposons. M^i représente $mode(D^i)$. Un point d'interrogation indique que le sens de variation de la protéine n'est pas identique partout dans le domaine.

Afin d'évaluer le surcoût en temps de calcul et en mémoire, lié à l'utilisation d'une partition plus fine, nous avons également fait figurer dans la table 7.1 les résultats obtenus avec la méthode proposée par de Jong *et al.* [70] (voir section 7.2.1). Cette comparaison montre, qu'en moyenne, le nombre d'états atteignables ne fait que doubler : on n'assiste pas à une augmentation exponentielle du nombre d'états atteignables. L'augmentation considérable du nombre total d'états dans l'espace des phases semble donc en général

contrebalancée par la spécification plus précise de l'état initial (un domaine de flot plutôt qu'un domaine de mode) et par l'augmentation de la finesse de l'analyse. De plus, bien que le nombre d'états atteignables augmente systématiquement, les temps de calcul ne sont pas toujours dégradés. Ceci provient d'optimisations algorithmiques plus poussées (voir partie III).

7.3 Validation du modèle

Dans la section précédente, nous avons montré comment obtenir des prédictions du comportement de différentes souches de *B. subtilis* placées dans des conditions défavorables. Ces prédictions nous permettent de tester la validité de notre modèle, en les comparant avec les données disponibles dans la littérature.

Nous utiliserons des données phénotypiques et des données d'expression génique. Les données phénotypiques concernent simplement la capacité d'une souche placée dans des conditions environnementales données à sporuler. Pour les données d'expression génétique, nous nous focalisons sur celles obtenues via deux techniques expérimentales dites à faible débit, donnant en général des résultats d'une bonne fiabilité : des analyses par Western blot et immunodétection, et des analyses par gènes rapporteurs. Les techniques de gènes rapporteurs sont en général utilisées pour mesurer l'activité des promoteurs et non pas les concentrations cellulaires des protéines. Toutefois, nous ferons ici l'hypothèse, sujette à caution, que les variations de l'activité des gènes rapporteurs sont représentatives des variations de la concentration de la protéine rapportée.

Toutes les propriétés de ce chapitre seront formalisées en CTL, et testées sur des graphes filtrés avec le model checker NuSMV. Des contraintes d'équité sont également ajoutées pour ne considérer que les exécutions qui atteignent effectivement un état d'équilibre (section 5.3). Les temps de calcul sont de l'ordre de la seconde pour l'importation des données (lecture du fichier et compilation du graphe de transitions en diagramme de décision binaire), et de l'ordre de la fraction de seconde pour la vérification effective des propriétés.

7.3.1 Validation par des données phénotypiques

Le premier type de propriété auquel nous nous intéressons concerne le phénotype, spo^+ ou spo^- , des cellules. Un phénotype spo^+ ou spo^- est dit observé si une partie non négligeable ($> 1\%$) de la population présente ce phénotype [67]. Cette information, facile à obtenir expérimentalement, est disponible pour le sauvage et pour la quasi-totalité des mutants identifiés. Dans la table 7.2 sont résumés les phénotypes observés pour un certain nombre de souches.

Nous avons supposé dans la section précédente que la cellule est engagée dans la voie de la sporulation dès que la concentration de σ^F , x_f , dépasse un certain seuil (θ_f). Ainsi, si l'on observe que certaines cellules sporulent, alors il doit exister au moins un chemin dans le graphe de transitions le long duquel à un certain moment x_f dépasse θ_f , ce qui s'exprime en CTL par $\mathbf{EF}(x_f > \theta_f)$. A l'inverse, si des cellules ne sporulent pas, alors il doit exister un chemin le long duquel x_f reste toujours inférieur à θ_f : $\mathbf{EG}(x_f < \theta_f)$. Naturellement, lorsqu'une population présente les deux phénotypes, les prédictions doivent satisfaire la conjonction des deux propriétés correspondantes, $\mathbf{EF}(x_f > \theta_f) \wedge \mathbf{EG}(x_f < \theta_f)$, traduisant

Modification génétique	Phénotype prédit	Phénotype observé	Référence
sauvage	spo^+, spo^-	spo^+, spo^-	[196]
mutant <i>spo0A</i>	spo^-	spo^-	[122]
mutant <i>sigH</i>	spo^-	spo^-	[122]
mutant <i>sinI</i>	spo^-	spo^-	[16]
mutant <i>kinA</i>	spo^-	spo^+, spo^-	[130]
mutant <i>sinR</i>	spo^+, spo^-	spo^+, spo^-	[93]
mutant <i>abrB</i>	spo^+, spo^-	spo^+, spo^-	[207]
mutant <i>spo0A abrB</i>	spo^-	spo^-	[180]
surexpression d' <i>hpr</i>	spo^-	spo^-	[179]

TAB. 7.2 – Phénotypes prédits obtenus par model checking et phénotypes observés, pour un certain nombre de souches de *B. subtilis* placées en conditions environnementales défavorables (présence du signal de sporulation). Voir de Jong *et al.* [67] pour une étude similaire.

l'existence d'au moins deux chemins, chacun satisfaisant une des deux propriétés ci-dessus.

$$\begin{aligned} \text{phénotype } spo^- : & \quad \mathbf{EG}(x_f < \theta_f), \\ \text{phénotype } spo^+, spo^- : & \quad \mathbf{EF}(x_f > \theta_f) \wedge \mathbf{EG}(x_f < \theta_f). \end{aligned}$$

Le model checker NuSMV est utilisé pour tester si les prédictions sont cohérentes avec les phénotypes observés pour les populations des différentes souches de *B. subtilis*. Ces résultats sont synthétisés dans la table 7.2. A une exception près (*kinA*), le phénotype prédit est cohérent avec les observations. Notamment, le phénotype spo^- de la souche qui surexprime *hpr* est correctement prédit, bien que les mécanismes qui aboutissent à ce phénomène soient encore incompris [197]. L'analyse de la séquence des événements qui conduisent au phénotype spo^- permet de proposer une explication pour ce phénomène. Le niveau élevé de Hpr, dû à la surexpression de *hpr*, inhibe en permanence l'expression des gènes de l'opéron *sin* à partir du promoteur P1 (promoteur de l'opéron), mais pas l'expression de *sinR* à partir du promoteur P3 (promoteur de *sinR* seul) (voir figure 7.2). Ceci empêche une synthèse, même transitoire, de SinI, qui chez le sauvage se dimérise avec SinR et relâche ainsi l'inhibition qu'exerce ce dernier sur la synthèse de Spo0A, nécessaire à l'engagement de la cellule dans la voie de la sporulation. Cet exemple illustre bien la capacité de la méthode à générer des prédictions non triviales et biologiquement pertinentes du comportement du système.

Par contre, les prédictions du modèle pour le mutant *kinA*, qui ne satisfont pas la propriété $\mathbf{EF}(x_f > \theta_f)$ (phénotype spo^+), sont en contradiction avec l'observation qu'une mutation de ce gène ne supprime pas totalement la capacité de la cellule à sporuler [130]. Ceci est lié à la présence d'autres kinases, en particulier KinB et KinC, ignorées dans notre modèle et jouant un rôle similaire à KinA [67]. Le modèle présenté peut facilement être étendu de façon à prendre en compte l'existence des autres kinases, ce qui résoud ce problème.

7.3.2 Validation par des données d'expression génique

Dans cette section, nous comparons les prédictions du modèle avec des données sur l'expression des gènes du réseau. L'expression temporelle de Spo0A est mesurée par des

techniques de Western blot et d'immunodétection dans [88]. Ces données, cohérentes avec [154], montrent l'existence d'une accumulation transitoire de Spo0A (voir figure 7.8). Les mêmes auteurs montrent de plus que cette accumulation est graduelle [89].

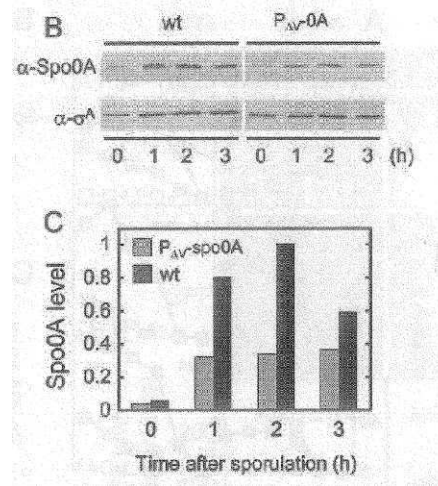


FIG. 7.8 – Profil d'expression de la protéine Spo0A [88]. La souche sauvage est notée wt. Les niveaux d'expression de Spo0A montrés en C ont été quantifiés à partir des blots B, normalisés par rapport aux niveaux de σ^A , puis par rapport au niveau d'expression maximum de Spo0A.

L'existence d'un chemin dans le graphe de transitions le long duquel la concentration de Spo0A augmente continuellement, jusqu'à atteindre un état dans lequel elle diminue, s'exprime en CTL selon

$$\mathbf{E}(\dot{x}_{sa} \geq 0 \mathbf{U} \dot{x}_{sa} < 0). \quad (7.2)$$

Cette formule exprime que la propriété $\dot{x}_{sa} \geq 0$ est vérifiée à chaque instant, et ce, jusqu'à un moment où la propriété $\dot{x}_{sa} < 0$ est vraie. Nous avons vérifié avec NuSMV que cette propriété est satisfaite par les prédictions.

De façon similaire, nous avons vérifié que les prédictions obtenues à partir du modèle sont également cohérentes avec le fait que la présence de glucose dans le milieu de culture, représentée par un signal de sporulation absent ($0 \leq u_s < \theta_s$), inhibe l'augmentation de la concentration de Spo0A [226] : dans ces conditions, la propriété (7.2) est fautive. Nous pouvons même vérifier que la propriété $\mathbf{AG}(\dot{x}_{sa} = 0)$ est vraie : la concentration de Spo0A ne varie pas.

D'autres données sont disponibles, concernant l'expression de *sigH* chez le sauvage et chez différents mutants. Dans [223], des expériences de fusion traductionnelle avec *lacZ*, le gène codant la β -galactosidase, montrent que l'activité β -galactosidase d'une souche sauvage contenant la construction *sigH-lacZ* augmente graduellement en fin de phase exponentielle, atteint un niveau élevé en début de sporulation (T_0) et y demeure après T_0 . La même étude a été réalisée chez plusieurs mutants. En particulier, l'activité β -galactosidase est trouvée constitutivement basse chez le mutant *spo0A* et constitutivement élevée chez les mutants *abrB* et *spo0A abrB* (voir figure 7.9 et [223]). Weir et collègues ont confirmé par Western blot et immunodétection que l'activité β -galactosidase mesurée est effectivement représentative de la concentration de la protéine σ^H [223].

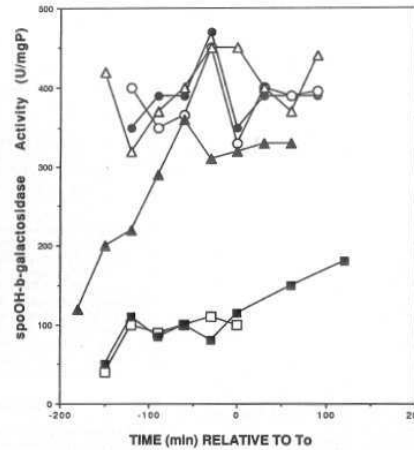


FIG. 7.9 – Activité β -galactosidase de la fusion traductionnelle *sigH-lacZ* dans différents contextes génétiques [223]. Les souches sont spo^+ (\blacktriangle), $spo0A$ (\square), $abrB$ (\triangle), et $spo0A\ abrB$ (\circ).

Nous souhaitons donc exprimer que chez la souche sauvage, la concentration de σ^H augmente graduellement puis demeure constante. L'existence d'un tel comportement du système peut s'exprimer en CTL comme :

$$\mathbf{E} (\dot{x}_h > 0 \mathbf{U} (\mathbf{E}\mathbf{G} \dot{x}_h = 0)). \quad (7.3)$$

L'expression constitutive (basse ou élevée) chez les mutants $spo0A$, $abrB$ et $spo0A\ abrB$ est simplement formalisée par $\mathbf{E}\mathbf{G} (\dot{x}_h = 0)$.

Lors du test de la propriété (7.3), NuSMV renvoie une réponse négative. L'analyse du contre-exemple donné par NuSMV montre que le problème provient du choix de l'état initial, qui ne satisfait pas $\dot{x}_h > 0$ (mais $\dot{x}_h = 0$). Les prédictions satisfont néanmoins la propriété

$$\mathbf{E}\mathbf{X} \mathbf{E} (\dot{x}_h > 0 \mathbf{U} (\mathbf{E}\mathbf{G} \dot{x}_h = 0)). \quad (7.4)$$

qui traduit l'existence d'un successeur de l'état initial qui satisfait la propriété (7.3). Cela ne remet donc pas en cause la validité du modèle, mais traduit simplement l'existence d'un certain délai, prédit mais non observé, avant que *sigH* ne soit exprimé. Pour les mutants, la vérification des propriétés ne pose pas de problème : les prédictions sont effectivement cohérentes avec les observations.

Dans [73], l'expression de plusieurs gènes impliqués dans la sporulation est mesurée dans des conditions identiques et avec des systèmes rapporteurs similaires, intégrés dans une même souche (fusion transcriptionnelle avec *lacZ*). Ces données qui mettent en parallèle l'expression de différents gènes sont particulièrement intéressantes pour notre étude. Nous considérons l'expression du gène *sinI* (figure 7.10(a)) et l'expression de l'opéron *spoIIA* contenant le gène *sigF* (figure 7.10(b)). Ces données montrent que pour les deux constructions, on observe une augmentation de l'activité β -galactosidase, qui est toutefois plus tardive pour *spoIIA*. Cette expression plus tardive de *spoIIA* est tout à fait cohérente avec les rôles attribués aux protéines SinI et σ^F : les protéines de l'opéron *sin*, dont SinI, jouent un rôle dans la transition entre la fin de phase exponentielle et le début de la sporulation, tandis que les protéines codées par l'opéron *spoIIA*, dont σ^F , interviennent

dans la formation de la préspore, lorsque la cellule est déjà engagée dans la voie de la sporulation [83].

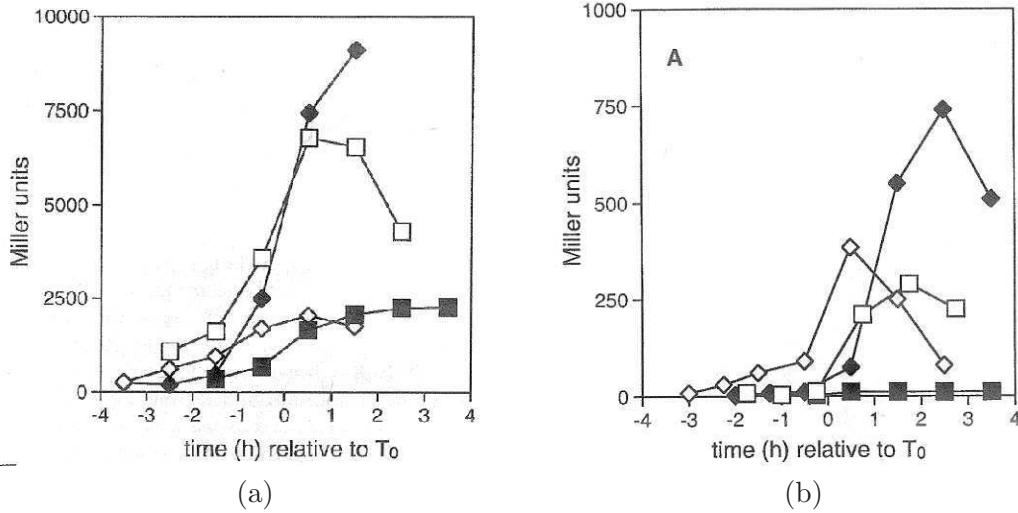


FIG. 7.10 – Activités β -galactosidase de (a) la fusion *sinI-lacZ* (\square) et (b) la fusion *spoIIA-lacZ* (\square) dans des cultures placées dans un milieu de sporulation [73]. L'instant 0 représente le début de la sporulation.

Nous souhaitons donc traduire l'existence d'un comportement du système tel qu'à un certain moment, seule la concentration de SinI augmente, puis que dans un deuxième temps, les concentrations des deux protéines augmentent. Cette propriété peut être traduite en CTL selon :

$$\mathbf{EF}(\dot{x}_{si} > 0 \wedge \dot{x}_f \leq 0 \wedge \mathbf{EF}(\dot{x}_{si} > 0 \wedge \dot{x}_f > 0)). \quad (7.5)$$

Il est à noter que dans (7.5), nous utilisons $\dot{x}_f \leq 0$ pour traduire que la concentration de σ^F n'augmente pas. Nous vérifions alors par model checking que les prédictions obtenues sont cohérentes avec cette observation.

Une étude plus précise des données présentées dans la figure 7.10 montre que, de plus, le pic d'activité β -galactosidase de *sinI-lacZ* (à $T_0+0,5$) est plus précoce que celui de *spoIIA-lacZ* (à T_0+2). Ainsi, après une période d'augmentation de l'activité β -galactosidase pour les deux gènes, il existe un moment où l'activité β -galactosidase de la fusion *sinI-lacZ* diminue alors que l'activité β -galactosidase de la fusion *spoIIA-lacZ* continue d'augmenter.

Nous traduisons donc que nous observons trois états successifs, tels que dans le premier, seule la concentration de SinI augmente, dans le second, les concentrations des deux protéines augmentent, et dans le troisième, la concentration de SinI diminue, tandis que celle de σ^F continue d'augmenter. Cette propriété peut être traduite en CTL selon :

$$\mathbf{EF}(\dot{x}_{si} > 0 \wedge \dot{x}_f \leq 0 \wedge \mathbf{EF}(\dot{x}_{si} > 0 \wedge \dot{x}_f > 0 \wedge \mathbf{EF}(\dot{x}_{si} < 0 \wedge \dot{x}_f > 0))). \quad (7.6)$$

Nous vérifions par model checking que cette propriété n'est pas satisfaite par les prédictions. De plus, nous pouvons montrer que, contrairement aux observations, la diminution de la concentration de SinI n'a jamais lieu avant celle de σ^F : $\mathbf{AG}(\neg(\dot{x}_{si} < 0 \wedge \dot{x}_f > 0))$.

Rappelons que les prédictions obtenues (formellement, le système de transitions LPM qualitatif) ne sont qu'une approximation du système original (formellement le système

LPM). Donc, il se peut que la contradiction obtenue soit non pas due à une contradiction entre le modèle et les données, mais une conséquence de l'abstraction utilisée. Pour répondre à cette question, nous pouvons utiliser les notions vues dans les sections 2.2.3 et 5.3.1 sur la vérification de systèmes abstraits, et en particulier le fait que si une formule de logique universelle est vraie pour le système abstrait, alors elle l'est également pour le système original. La négation de (7.6),

$$\mathbf{AG}(\dot{x}_{si} \leq 0 \vee \dot{x}_f > 0 \vee \mathbf{AG}(\dot{x}_{si} \leq 0 \vee \dot{x}_f \leq 0 \vee \mathbf{AG}(\dot{x}_{si} \geq 0 \vee \dot{x}_f \leq 0))). \quad (7.7)$$

est une formule de logique universelle (dans $\forall\text{CTL}$), qui est vraie (puisque nous avons vu que sa négation était fausse). Nous en concluons que la contradiction trouvée entre les prédictions et l'observation reflète effectivement une contradiction entre le modèle et les données. Une révision du modèle est alors nécessaire pour prendre en compte ce phénomène. Une discussion de ce problème, et des extensions du modèle, sont proposées dans de Jong *et al.* [67].

La dernière propriété que nous nous proposons de tester concerne l'expression de *hpr*. Perego et Hoch [179] montrent par des expériences de fusion traductionnelle, que l'activité β -galactosidase d'une souche portant la construction *hpr-lacZ* augmente jusqu'à atteindre un niveau élevé en début de phase stationnaire, puis reste à ce niveau élevé (figure 7.11).¹

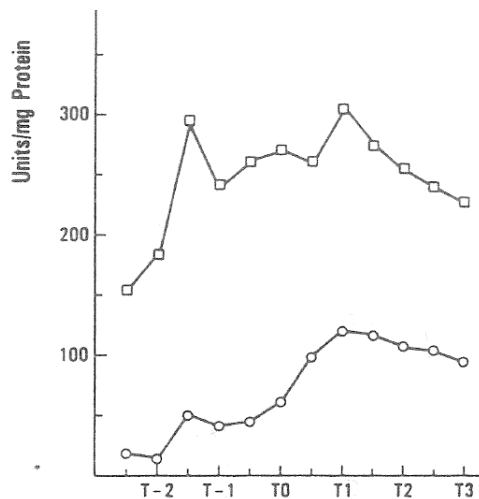


FIG. 7.11 – Activité β -galactosidase d'une fusion traductionnelle *hpr-lacZ* (\circ) dans une culture placée dans un milieu de sporulation [179]. L'instant T_0 représente le début de la sporulation.

Sous l'hypothèse que l'activité β -galactosidase mesurée représente la concentration cellulaire de Hpr, nous pouvons formaliser en CTL l'observation précédente selon :

$$\mathbf{E}(\dot{x}_{hr} > 0 \mathbf{U}(\mathbf{EG} \dot{x}_{hr} = 0)). \quad (7.8)$$

Lorsque nous testons cette propriété avec NuSMV, nous obtenons une réponse négative du model checker. Pire encore, le long de tous les comportements prédits, la concentration

¹« In the wild-type strain, the specific activity of β -galactosidase expressed from the *hpr-lacZ* fusion increased in proportion to the growth curve, reached a maximum level, at the early stationary phase, and remained at the same level during the stationary phase » [179].

de Hpr diminue jusqu'à atteindre une valeur d'équilibre : $\mathbf{A}(\dot{x}_{hr} \leq 0 \mathbf{U}(\mathbf{AG} \dot{x}_{hr} = 0))$. Comme pour la propriété précédente (7.6), la négation de (7.8) est dans $\forall\text{CTL}$ et est vraie, ce qui montre que la contradiction trouvée entre les prédictions et l'observation reflète effectivement une contradiction entre le modèle et les données.

Il est possible de proposer des modifications du modèle. Par exemple, nous pouvons supposer que des régulateurs autres que Spo0A et AbrB interviennent dans le contrôle de l'expression de *hpr*. Il n'existe toutefois pas, à notre connaissance, de données qui étayent cette hypothèse. De plus, l'observation publiée par Perego et Hoch [179] semble être en contradiction avec le rôle généralement proposé pour Hpr, celui d'être, entre autres, un inhibiteur de la sporulation, puisque une surexpression d'Hpr entraîne un blocage de la sporulation [179], et en contradiction avec le fonctionnement proposé par Strauch et Hoch de cette partie du réseau [208] :

« Upon entry into the transition-state, Spo0A~P repression of AbrB synthesis leads to the lowering of *hpr* expression ».

Pour résoudre ces contradictions, il semble nécessaire de réaliser des expériences complémentaires, et notamment de mesurer l'évolution de la concentration de la protéine Hpr lors de l'entrée en sporulation.

Dans ce chapitre, nous avons appliqué la méthode présentée dans la partie II à la validation d'un modèle de l'initiation de la sporulation chez *B. subtilis* en comparant les prédictions obtenues avec un ensemble de données expérimentales disponibles dans la littérature.

Les résultats obtenus illustrent l'applicabilité de l'approche proposée à la validation de modèles grands et complexes. Premièrement, nous montrons que l'utilisation d'une méthode d'analyse relativement fine, permettant notamment de prédire les sens de variation des concentrations des protéines du réseau, évite de réaliser une comparaison trop conservative entre prédictions et observations, sans compromettre les capacités de mise à l'échelle. Deuxièmement, nous montrons que l'utilisation de model checking permet de formaliser en logique temporelle un large éventail de propriétés observées sur les systèmes et de tester efficacement leur cohérence avec les prédictions du modèle.

Egalement, les résultats obtenus dans ce chapitre montrent qu'il existe globalement un bon accord entre les prédictions obtenues à partir du modèle proposé par de Jong *et al.* [67] et les données expérimentales disponibles dans la littérature. Quelques incohérences entre les prédictions du modèle et des observations ont toutefois été identifiées, qui nous ont conduit soit à suggérer des extensions du modèle, soit à proposer la réalisation d'expériences complémentaires.

Analyse de la réponse au stress nutritionnel chez *E. coli*

Dans ce chapitre, nous nous intéressons à un autre exemple d'adaptation bactérien, la réponse au stress nutritionnel chez *Escherichia coli*. Comme chez *B. subtilis*, ce phénomène a fait l'objet de plusieurs décennies d'études. Pourtant, malgré l'abondance d'information disponible sur les différents composants du système, il n'existe pas de vision globale de son fonctionnement. Également, la quasi-absence de valeurs numériques pour les valeurs des paramètres et des concentrations rend difficilement applicables les méthodes traditionnelles de modélisation. Pour tenter de mieux comprendre ce phénomène, un modèle LPM qualitatif a été construit et analysé par Ropers *et al.* [192].

Dans la première section, nous présentons plus en détail le phénomène de la réponse au stress nutritionnel chez *E. coli*, ainsi que le modèle proposé dans [192]. Puis, nous utilisons la méthode d'analyse qualitative présentée dans la partie II pour prédire la réponse des cellules en phase exponentielle confrontées à un stress nutritionnel (entrée en phase stationnaire), et la réponse des cellules en phase stationnaire lors d'un apport soudain de glucose (réentrée en phase exponentielle). Les prédictions obtenues sont alors utilisées, à la section 8.3, pour tester la validité du modèle, en les comparant avec des données de la littérature. Comme précédemment, les propriétés sont formalisées en logique temporelle et testées par model checking.

Une synthèse des résultats présentés dans ce chapitre est proposée dans [24].

8.1 Modélisation de la réponse au stress nutritionnel

8.1.1 Réponse au stress nutritionnel chez *E. coli*

Dans leur environnement naturel, des bactéries comme *E. coli* rencontrent rarement des conditions de croissance continue. Tant que des nutriments sont disponibles, les cellules d'*E. coli* croissent et se divisent rapidement, ce qui conduit à une augmentation exponentielle de leur biomasse, un état appelé *phase exponentielle*. Lorsqu'un nutriment essentiel vient à manquer, les bactéries ne peuvent plus maintenir des vitesses de croissance élevées et la population entre dans un état de croissance ralentie voire nulle, appelé *phase stationnaire*. La transition de la phase exponentielle à la phase stationnaire fait intervenir de profonds changements physiologiques, notamment de la morphologie, du métabolisme cellulaire ou encore de l'expression des gènes [127]. Ces changements rendent la cellule capable de survivre à des périodes prolongées de privation et de résister à des stress multiples. Cette *réponse au stress nutritionnel* est réversible et la croissance peut reprendre dès que des nutriments sont à nouveau disponibles.

Les transitions, de la phase exponentielle à la phase stationnaire, et inversement, sont contrôlées par un réseau de régulation génique complexe, réalisant notamment l'intégration de multiples signaux environnementaux [156, 225]. Bien que l'objectif à long terme soit d'analyser l'ensemble de ce grand réseau, l'étude proposée se focalise, dans un premier temps, sur une partie de celui-ci, responsable plus particulièrement de la réponse au *manque de carbone*.

8.1.2 Réseau de régulation génique contrôlant la réponse au manque de carbone

Parmi les gènes qui contrôlent la réponse au manque de carbone, deux régulateurs globaux jouent un rôle clé : Fis et CRP [219, 194]. Les régulateurs globaux sont des facteurs de transcription ayant des effets pléiotropiques et contrôlant l'expression d'un très grand nombre de gènes (jusqu'à 51% des gènes d'*E. coli*) [156]. Nous considérerons par la suite la topologie de l'ADN comme un autre régulateur global, car l'expression d'un grand nombre de gènes dépend également du niveau de superenroulement de l'ADN [181]. Le réseau que nous considérons inclut également une entrée (signal de manque de carbone) et une sortie représentant de l'état de croissance cellulaire (concentration des ARNs stables). Les différents éléments, et leurs interactions, formant ce réseau sont représentés dans la figure 8.1 et sont succinctement décrits dans la suite de cette section. Une description plus complète de ce réseau est donnée par Ropers *et al.* dans [192].

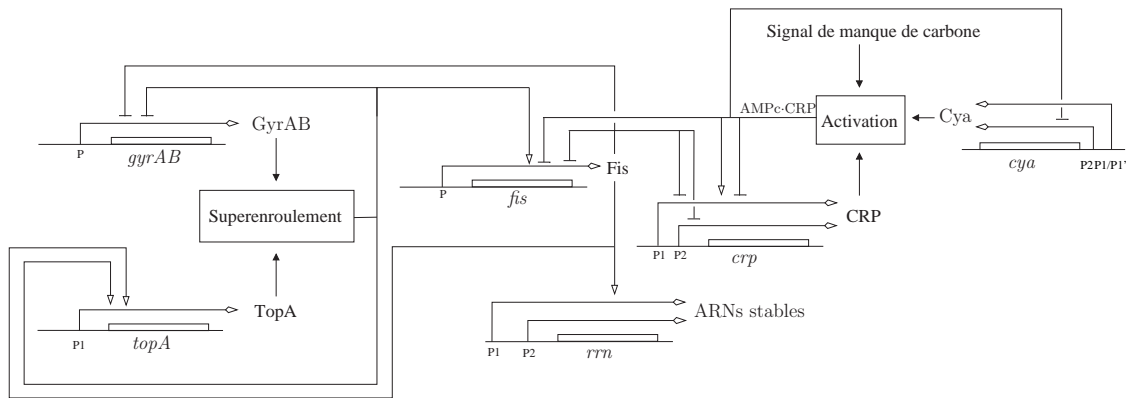


FIG. 8.1 – Gènes, protéines et interactions clés, impliqués dans la réponse au manque de carbone chez *Escherichia coli*. Les conventions graphiques sont données dans la figure 7.1. Les contenus des boîtes appelées Superenroulement et Activation sont détaillés dans la figure 8.2. [192].

Le manque de carbone est détecté par le système phosphotransférase (PTS), ce qui entraîne alors l'activation de l'adénylate cyclase (Cya), causant la production d'AMPc à partir de l'ATP [194]. L'AMPc est un petit métabolite se liant à CRP [194]. Nous simplifions la voie de transduction du signal de manque de carbone en considérant que le signal active directement Cya (figure 8.2(a)).

CRP, lorsqu'il est lié à l'AMPc, contrôle, entre autre, l'expression de plusieurs régulateurs globaux, dont Fis et CRP lui-même [156]. De façon similaire, Fis contrôle sa propre expression et celle de CRP [104]. Fis et CRP répriment réciproquement leur expression [104]. L'expression de Fis, en plus d'être contrôlée par CRP et par Fis lui-même, dépend de la topologie de l'ADN [219]. Le niveau de superenroulement de l'ADN dépend principalement de l'activité antagoniste de deux topoisomérases, GyrAB et TopA : GyrAB augmente le superenroulement de l'ADN, tandis que TopA le diminue [74]. Le niveau de superenroulement est contrôlé de façon homéostatique : une augmentation de la concentration d'une topoisomérase entraîne une modification de l'expression génique aboutissant à la production de l'autre topoisomérase, rétablissant ainsi le niveau de superenroulement physiologique [74]. En plus de ces contrôles homéostatiques, la topologie de l'ADN est éga-

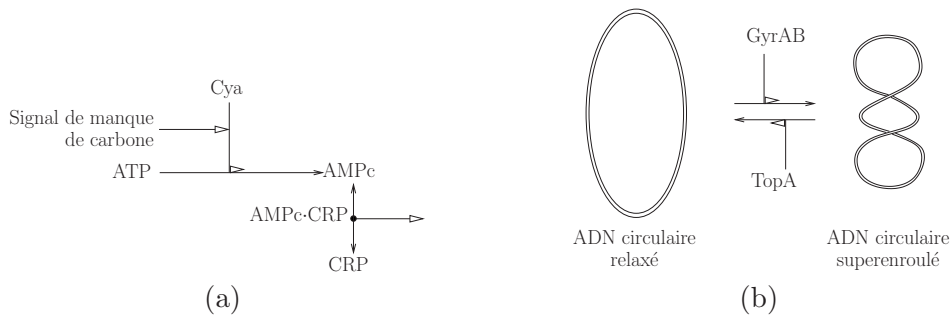


FIG. 8.2 – (a) Contrôle de la formation du complexe AMPc-CRP. Le signal de manque de carbone active l'adénylate cyclase (Cya), entraînant ainsi la production d'AMPc à partir d'ATP. L'AMPc se lie au régulateur global CRP en formant ainsi le complexe d'AMPc-CRP. (b) Contrôle de la topologie de l'ADN. Le niveau de superenroulement de l'ADN dépend essentiellement du ratio des activités de GyrAB, qui sur-enroule l'ADN, et de TopA, qui au contraire relaxe l'ADN. [192].

lement contrôlée par Fis, via le contrôle de la production des deux topoisomérases [219]. En réalité, deux gènes, *gyrA* et *gyrB*, sont nécessaires à la production de l'hétérodimère GyrAB. Etant donné que leur régulation est similaire, nous considérerons que *gyrA* et *gyrB* ne forment qu'un gène, *gyrAB*, codant la protéine GyrAB.

Enfin, le réseau représenté figure 8.1 inclut les ARN stables, codés par les opérons *rrn* et fortement stimulés par Fis [12]. Etant donné que les besoins des cellules en ARN stables pour traduire les ARNm en protéines sont très importants en phase de croissance et très modestes en phase stationnaire [225], nous considérerons que la concentration en ARN stables est représentative de l'état de croissance de la cellule. Plus précisément, nous considérerons que les cellules sont en phase stationnaire lorsque la concentration des ARN stables est basse, et en phase exponentielle sinon.

La réponse de la cellule à la présence ou à l'absence d'une source de carbone émerge des interactions entre les différents composants du réseau. Etant donné la taille (nombre de composants) et la complexité (présence de boucles de rétroaction) de ce réseau, la compréhension intuitive de ce phénomène est difficile. Aussi, afin de pouvoir comprendre comment les réponses de la cellule émergent des interactions entre composants du réseau, un modèle LPM qualitatif a été construit [192].

8.1.3 Modèle LPM du réseau contrôlant la réponse au manque de carbone

A partir de l'abondante littérature et des bases de données EcoCyc [132] et RegulonDB [195], un modèle LPM qualitatif a été construit, incluant une variable d'état pour chacun des gènes codant CRP, Cya, Fis, GyrAB, TopA, ou les ARN stables, ainsi qu'une variable d'entrée représentant la présence ou l'absence d'un signal de manque de carbone [192].

Ce modèle, représenté figure 8.3, est constitué de 7 équations différentielles LPM et d'autant d'ensembles de contraintes d'inégalité sur les paramètres, représentant en tout 36 inégalités. A quelques exceptions près, il a été possible de spécifier les fonctions de régulation et les contraintes d'inégalité sur les paramètres entièrement à partir des données disponibles dans la littérature expérimentale. L'ensemble des étapes de la construction du

$\dot{u}_s = 0$ $0 < \theta_s < \max_s$
$\dot{x}_{crp} = \kappa_{crp}^1 + \kappa_{crp}^2 s^-(x_{fis}, \theta_{fis}^2) s^+(x_{crp}, \theta_{crp}^1) s^+(x_{cya}, \theta_{cya}^1) s^+(u_s, \theta_s) + \kappa_{crp}^3 s^-(x_{fis}, \theta_{fis}^1) - \gamma_{crp} x_{crp}$ $0 < \theta_{crp}^1 < \kappa_{crp}^1 / \gamma_{crp} < (\kappa_{crp}^1 + \kappa_{crp}^2) / \gamma_{crp} < \theta_{crp}^2 < \theta_{crp}^3 < (\kappa_{crp}^1 + \kappa_{crp}^3) / \gamma_{crp}$ $< (\kappa_{crp}^1 + \kappa_{crp}^2 + \kappa_{crp}^3) / \gamma_{crp} < \max_{crp}$
$\dot{x}_{cya} = \kappa_{cya}^1 + \kappa_{cya}^2 (1 - s^+(x_{crp}, \theta_{crp}^3) s^+(x_{cya}, \theta_{cya}^3) s^+(u_s, \theta_s)) - \gamma_{cya} x_{cya}$ $0 < \theta_{cya}^1 < \kappa_{cya}^1 / \gamma_{cya} < \theta_{cya}^2 < \theta_{cya}^3 < (\kappa_{cya}^1 + \kappa_{cya}^2) / \gamma_{cya} < \max_{cya}$
$\dot{x}_{fis} = \kappa_{fis}^1 (1 - s^+(x_{crp}, \theta_{crp}^1) s^+(x_{cya}, \theta_{cya}^1) s^+(u_s, \theta_s)) s^-(x_{fis}, \theta_{fis}^5)$ $+ \kappa_{fis}^2 s^+(x_{gyrAB}, \theta_{gyrAB}^1) s^-(x_{topA}, \theta_{topA}^2) s^-(x_{fis}, \theta_{fis}^5) (1 - s^+(x_{crp}, \theta_{crp}^1) s^+(x_{cya}, \theta_{cya}^1) s^+(u_s, \theta_s))$ $- \gamma_{fis} x_{fis}$ $0 < \theta_{fis}^1 < \kappa_{fis}^1 / \gamma_{fis} < \theta_{fis}^2 < \theta_{fis}^3 < \theta_{fis}^4 < \theta_{fis}^5 < (\kappa_{fis}^1 + \kappa_{fis}^2) / \gamma_{fis} < \max_{fis}$
$\dot{x}_{gyrAB} = \kappa_{gyrAB}^1 + \kappa_{gyrAB}^2 (1 - s^+(x_{gyrAB}, \theta_{gyrAB}^3) s^-(x_{topA}, \theta_{topA}^1) s^-(x_{fis}, \theta_{fis}^4)) - \gamma_{gyrAB} x_{gyrAB}$ $0 < \kappa_{gyrAB}^1 / \gamma_{gyrAB} < \theta_{gyrAB}^1 < \theta_{gyrAB}^2 < \theta_{gyrAB}^3 < (\kappa_{gyrAB}^1 + \kappa_{gyrAB}^2) / \gamma_{gyrAB} < \max_{gyrAB}$
$\dot{x}_{topA} = \kappa_{topA}^1 + \kappa_{topA}^2 s^+(x_{gyrAB}, \theta_{gyrAB}^3) s^-(x_{topA}, \theta_{topA}^1) s^+(x_{fis}, \theta_{fis}^4) - \gamma_{topA} x_{topA}$ $0 < \kappa_{topA}^1 / \gamma_{topA} < \theta_{topA}^1 < \theta_{topA}^2 < \theta_{topA}^3 < (\kappa_{topA}^1 + \kappa_{topA}^2) / \gamma_{topA} < \max_{topA}$
$\dot{x}_{rrn} = \kappa_{rrn}^1 + \kappa_{rrn}^2 s^+(x_{fis}, \theta_{fis}^3) - \gamma_{rrn} x_{rrn}$ $0 < \kappa_{rrn}^1 / \gamma_{rrn} < \theta_{rrn} < (\kappa_{rrn}^1 + \kappa_{rrn}^2) / \gamma_{rrn} < \max_{rrn}$

FIG. 8.3 – Equations différentielles LPM et contraintes d’inégalité sur les paramètres pour le réseau de la réponse au stress nutritionnel représenté figure 8.1. Le signal de manque de carbone est représenté par u_s . [192].

modèle est décrit en détail dans [192].¹

8.2 Analyse qualitative

Dans cette section, nous utilisons la méthode d’analyse qualitative présentée dans la partie II pour simuler la réponse de cellules en phase exponentielle confrontées à un manque de carbone, et la réponse de cellules en phase stationnaire lors d’un apport soudain d’une source de carbone (glucose). Les prédictions obtenues seront analysées par model checking dans la section suivante. Mentionnons que les résultats donnés au chapitre 7 sur la comparaison entre la méthode que nous utilisons ici et celle proposée par de Jong *et al.* dans [70] demeurent valables ici : les résultats obtenus sont souvent plus précis, ce gain de précision étant en général essentiel pour éviter une validation trop conservatrice, au prix d’une augmentation modérée du nombre des états atteignables (en moyenne, doublement

¹La distribution actuelle de GNA inclut un modèle légèrement différent, plus récent que celui étudié dans ce chapitre. Ce dernier est toutefois disponible sur demande. Contact : delphine.ropers@inrialpes.fr.

du nombre des états).

8.2.1 Simulation de l'entrée en phase stationnaire

En conditions favorables, c'est-à-dire en l'absence du signal de manque de carbone, ($0 \leq u_s < \theta_s$), le système possède un unique état d'équilibre, qui est singulier :

$$\begin{aligned} x_{crp} &= \kappa_{crp}^1 / \gamma_{crp}, & x_{cya} &= (\kappa_{cya}^1 + \kappa_{cya}^2) / \gamma_{cya}, \\ x_{fis} &= \theta_{fis}^4, & x_{gyrAB} &= \theta_{gyrAB}^2, \\ x_{topA} &= \kappa_{topA}^1 / \gamma_{topA}, & x_{rrn} &= (\kappa_{rrn}^1 + \kappa_{rrn}^2) / \gamma_{rrn}. \end{aligned} \quad (8.1)$$

L'état d'équilibre trouvé est cohérent avec l'état physiologique des cellules en phase de croissance exponentielle. On vérifie tout d'abord que la concentration des ARN stables est élevée ($x_{rrn} > \theta_{rrn}$). De plus, les valeurs des variables sont également en accord avec le fait que chez *E. coli*, la concentration de CRP est basse [129], alors que les concentrations de Cya [194] et de Fis [18] sont élevées, dans des cellules en phase exponentielle. Enfin, une concentration de GyrAB assez élevée et de TopA basse sont en accord avec le niveau de superenroulement élevé observé [17].

A partir de cet état, nous simulons un épuisement de la source de carbone du milieu en commutant le signal de manque de carbone : $\theta_s < u_s \leq max_s$. Nous utilisons la nouvelle version de GNA (partie III) pour calculer l'ensemble des états atteignables. Le graphe de transitions est alors calculé en moins d'une seconde sur un PC (800MHz, 256Mo) et contient 66 états, dont 27 seulement sont persistants. Ce graphe est représenté figure 8.4. Le graphe contient un unique état d'équilibre, atteignable à partir de tous les états du graphe :

$$\begin{aligned} \theta_s &< u_s < max_s, & x_{cya} &= \theta_{cya}^3, \\ x_{crp} &= (\kappa_{crp}^1 + \kappa_{crp}^2 + \kappa_{crp}^3) / \gamma_{crp}, & x_{gyrAB} &= \theta_{gyrAB}^3, \\ x_{fis} &= 0, & x_{rrn} &= \kappa_{rrn}^1 / \gamma_{rrn}. \\ x_{topA} &= \kappa_{topA}^1 / \gamma_{topA}, \end{aligned} \quad (8.2)$$

L'étude détaillée des résultats de simulation fait l'objet de la section 8.3, mais nous pouvons dès à présent remarquer que dans l'état d'équilibre atteint par le système, la concentration des ARN stables est basse ($x_{rrn} < \theta_{rrn}$). Nous en concluons que cet état d'équilibre correspond à des cellules en phase stationnaire. Ainsi, le modèle rend déjà compte du fait que confrontées à un manque de carbone, des cellules en phase exponentielle entrent en phase stationnaire. Un chemin du graphe de transitions correspondant à une exécution typique du système allant de l'état initial instantané D^1 à l'état d'équilibre D^{28} est représenté figure 8.5. Dans ce chemin, les états instantanés autres que D^1 ne sont pas représentés.

8.2.2 Simulation de la réentrée en phase exponentielle

Afin de simuler la réponse de cellules en phase stationnaire à l'apport soudain d'une source de carbone (glucose), nous choisissons comme état initial du système l'état d'équilibre (8.2), identifié plus haut comme représentant des cellules en phase stationnaire, et nous commutons à nouveau le signal de manque de carbone, de façon à mimer une disponibilité soudaine de glucose : $0 \leq u_s < \theta_s$. Mentionnons toutefois que par rapport à

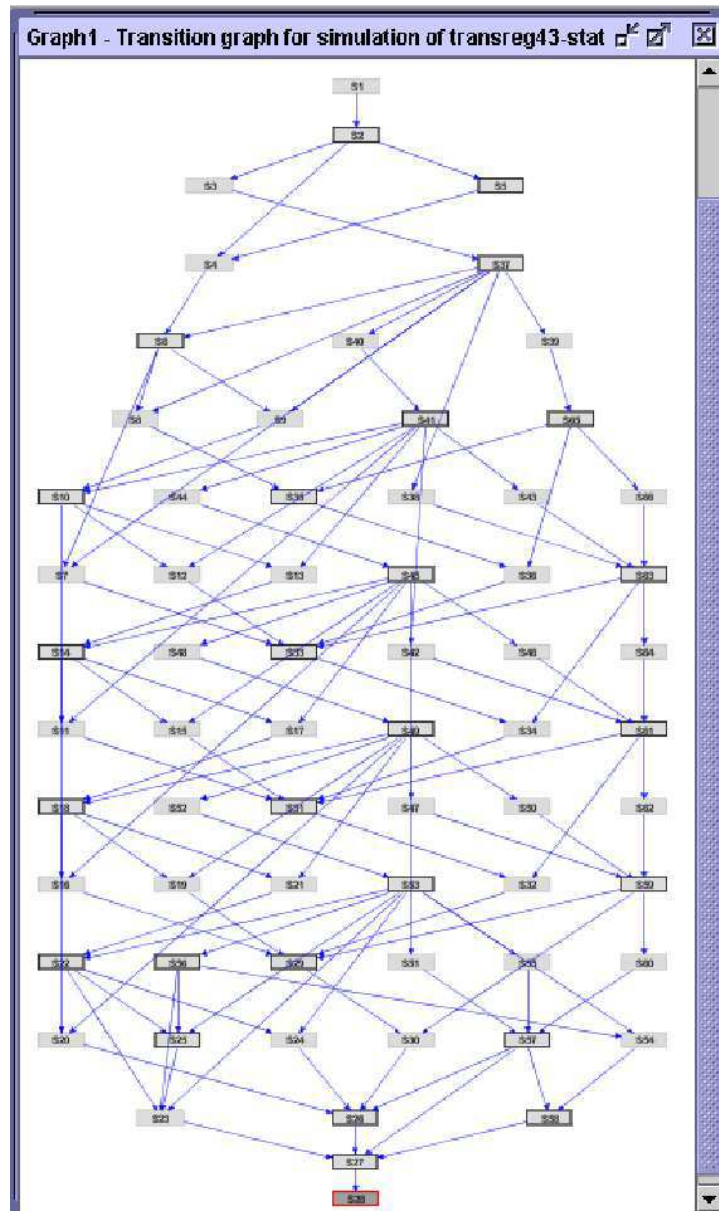


FIG. 8.4 – Graphe de transitions obtenu lors de la simulation d’une transition de la phase exponentielle à la phase stationnaire causée par un épuisement du milieu en source de carbone, chez *E. coli*.

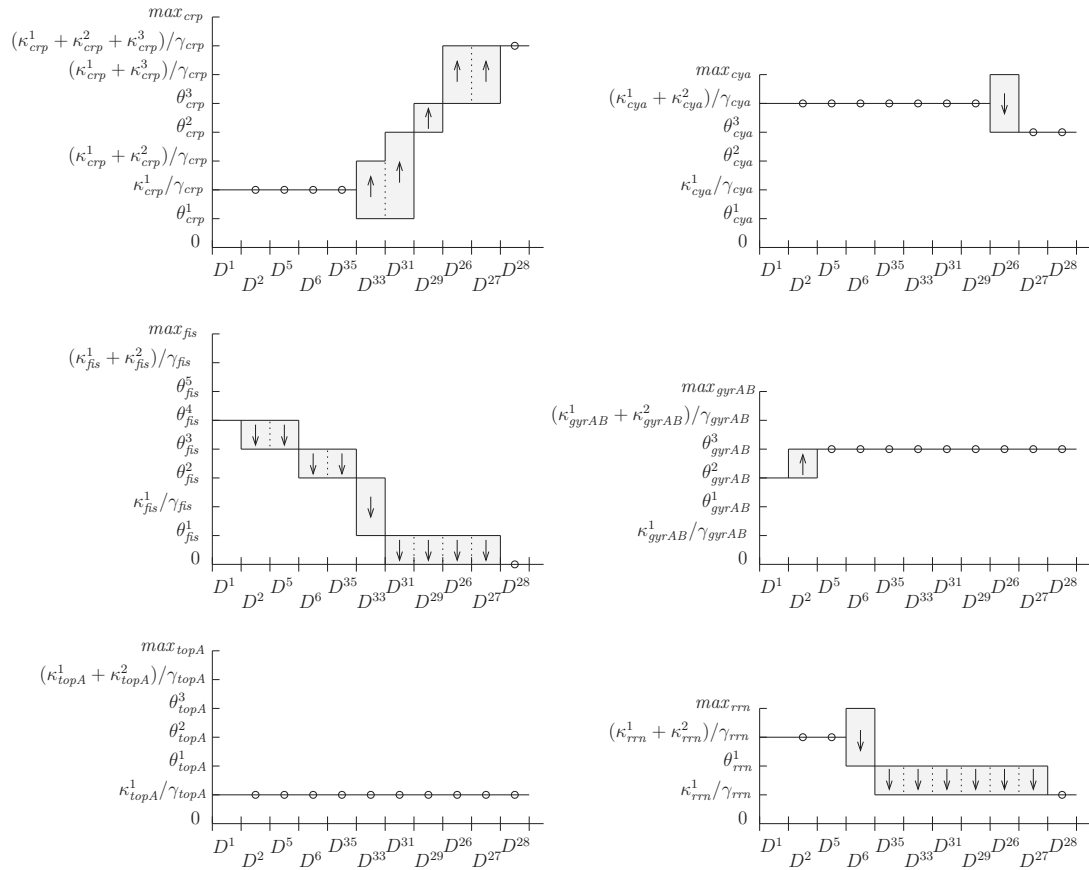


FIG. 8.5 – Evolution temporelle de la concentration des protéines et des ARN stables du réseau de la réponse au stress nutritionnel le long d'un chemin du graphe de transitions, illustrant la transition de la phase exponentielle à la phase stationnaire causée par un manque de carbone. La représentation utilisée est décrite dans la légende de la figure 7.6.

(8.2), la concentration de la protéine GyrAB est modifiée dans l'état initial choisi. La justification de cette correction, nécessaire pour rendre compte du niveau physiologique du superenroulement des cellules en phase stationnaire, sera donnée dans la section suivante.

$$\begin{aligned}
\theta_s &< u_s < max_s, \\
x_{crp} &= (\kappa_{crp}^1 + \kappa_{crp}^2 + \kappa_{crp}^3) / \gamma_{crp}, & x_{cya} &= \theta_{cya}^3, \\
x_{fis} &= 0, & x_{gyrAB} &= \kappa_{gyrAB}^1 / \gamma_{gyrAB}, \\
x_{topA} &= \kappa_{topA}^1 / \gamma_{topA}, & x_{rrn} &= \kappa_{rrn}^1 / \gamma_{rrn}.
\end{aligned} \tag{8.3}$$

En une seconde environ, le graphe de transitions est calculé. Il contient 1143 états, dont 202 seulement sont persistants. Il contient également trois composantes fortement connexes, parfois appelées aussi *cycles qualitatifs* [70]. Chaque composante contient plus de 200 états et possède une structure complexe. Un fragment du graphe est représenté figure 8.6. L'interprétation biologique de ces prédictions est difficile et nécessite une analyse détaillée de ce grand graphe et de ses composantes. Nous verrons dans la section suivante, que le model checking, en plus de permettre de tester efficacement la validité des modèles en testant si les prédictions obtenues sont cohérentes avec les données expérimentales, peut également fournir une aide précieuse pour faciliter l'analyse des grands graphes et ainsi l'interprétation biologique des résultats de simulation.

8.3 Interprétation et validation du modèle

Dans la première section, nous utilisons les possibilités offertes par le model checking pour l'exploration des propriétés des grands graphes, afin de faciliter l'interprétation biologique des résultats de simulation complexes obtenus précédemment (section 8.2.2). Dans la seconde section, nous utilisons le model checking pour tester la validité du modèle en confrontant les résultats de simulation obtenus avec les propriétés observées du système.

8.3.1 Interprétation biologique des résultats de simulation

Pour tenter de mieux comprendre la structure du graphe obtenu à la section 8.2.2, nous définissons trois propositions supplémentaires : $inSCC^1$, $inSCC^2$ et $inSCC^3$. La proposition $inSCC^i$ est vraie pour tous les états de la i ème composante fortement connexe, SCC^i (Strongly Connected Component), et pour ceux-ci seulement. Formellement,

$$inSCC^i =_{def} \bigvee_{D^j \in SCC^i} inD^j, \quad i \in \{1, 2, 3\},$$

avec inD^j , la conjonction de toutes les propositions atomiques vraies dans l'état D^j (voir section 5.2). En fait, ces propositions sont systématiquement définies lors de l'export des graphes par GNA, dès que le graphe contient des composantes fortement connexes.

Les composantes fortement connexes dites *terminales*, c'est-à-dire n'ayant aucune transition sortante, sont particulièrement intéressantes, car la région de l'espace des phases correspondante est positivement invariante : une fois cette région atteinte, le système ne peut plus en sortir. Ceci nous donne naturellement des informations précieuses sur la dynamique du système. Pour identifier ces composantes terminales, nous testons pour les différentes composantes identifiées si la propriété suivante est satisfaite.

$$\mathbf{AG}(inSCC^i \rightarrow \mathbf{AG}inSCC^i), \quad i \in \{1, 2, 3\}.$$

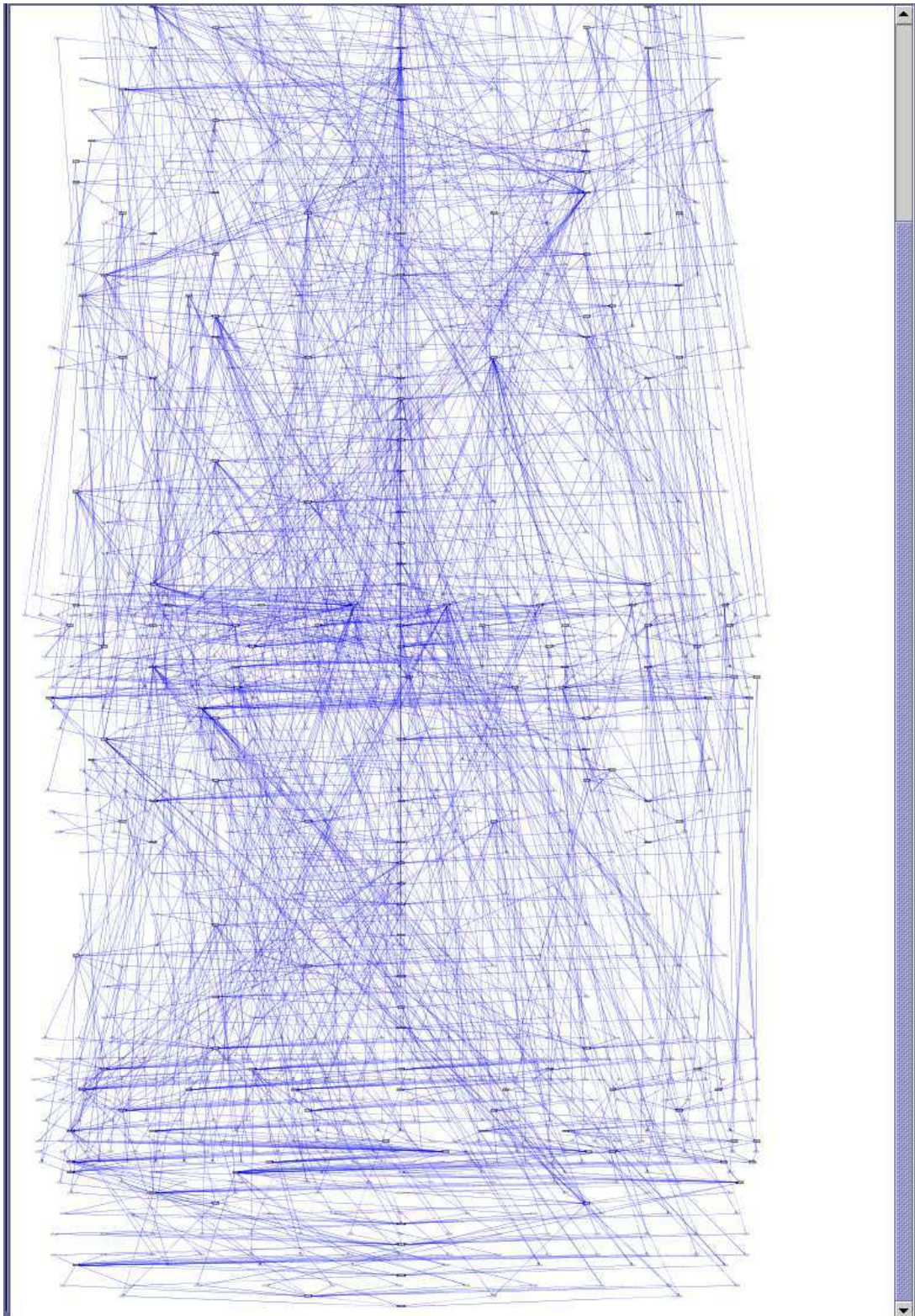


FIG. 8.6 – Graphe de transitions obtenu lors de la simulation d'une transition de la phase stationnaire à la phase exponentielle consecutive à un apport soudain d'une source de carbone, chez *E. coli*.

Cette propriété exprime que si le système est dans une composante donnée, alors nécessairement, il y reste toujours. Nous montrons alors avec NuSMV que seule la troisième composante fortement connexe satisfait cette propriété. Cette propriété est vérifiée en une fraction de seconde.

La présence de composantes fortement connexes dans le graphe de transitions correspond biologiquement à la présence possible d'oscillations dans les concentrations des protéines du réseau. Par model checking, nous pouvons caractériser précisément les oscillations qui ont lieu dans les composantes fortement connexes. Effectivement, nous avons vu à la section 5.2.2 que l'on pouvait traduire en μ -calcul l'existence d'un comportement cyclique dans lequel deux états alternent, l'un vérifiant une propriété p_1 , l'autre vérifiant une propriété p_2 par la formule suivante :

$$\begin{aligned} & \nu X.((p_1 \rightarrow \mu Y_1.((p_2 \wedge X) \vee \diamond Y_1)) \\ & \wedge (p_2 \rightarrow \mu Y_2.((p_1 \wedge X) \vee \diamond Y_2))). \end{aligned}$$

Par cette méthode, nous avons ainsi pu montrer (<2 s) avec XTL, un composant de la boîte à outils CADP, l'existence dans la composante fortement connexe terminale d'un cycle passant par deux états, l'un dans lequel la concentration de GyrAB est élevée et celle de Fis basse (état satisfaisant $p_1 =_{def} inSCC^3 \wedge x_{gyrAB} > \theta_{gyrAB}^2 \wedge x_{fis} < \theta_{fis}^2$), et l'autre dans lequel la concentration de GyrAB est basse et celle de Fis élevée (état satisfaisant $p_2 =_{def} inSCC^3 \wedge x_{gyrAB} < \theta_{gyrAB}^1 \wedge x_{fis} > \theta_{fis}^4$). Autrement dit, GNA prédit l'existence possible d'oscillations dans les concentrations de GyrAB et de Fis. Remarquons, comme expliqué dans la section 5.2.2, que cette propriété ne peut être exprimée en CTL, et ne peut être testée avec NuSMV. Ceci explique l'utilisation du μ -calcul et de la boîte à outils CADP.

L'information disponible dans le graphe n'est pas suffisante pour déterminer si ces oscillations correspondent dans l'espace des phases à un cycle limite ou à des trajectoires spiralant vers un point d'équilibre. Une analyse plus détaillée montre que pour toutes les valeurs de paramètres satisfaisant les contraintes d'inégalité données figure 8.3, les oscillations correspondent à des trajectoires qui spiralent vers un point d'équilibre (Jean-Luc Gouzé, communication personnelle). De plus, ce point d'équilibre correspond au domaine donné en (8.1), qui comme indiqué à la section 8.2.1, représente des cellules en phase de croissance exponentielle (concentration basse des ARN stables).

En résumé, le modèle rend bien compte du fait que des cellules en phase stationnaire mises en présence soudaine d'une source de carbone entrent en phase exponentielle. De plus, le modèle prédit que lors de cette réentrée en phase exponentielle, des oscillations amorties dans les concentrations de GyrAB et de Fis peuvent avoir lieu.

8.3.2 Validation du modèle

Comme dit en introduction, le phénomène de la réponse au stress nutritionnel a fait l'objet de plusieurs décennies d'études. Malgré cela, nous avons trouvé dans la littérature relativement peu de données sur la dynamique du système, l'essentiel des études portant sur l'analyse de l'état d'équilibre du système, consécutivement à des perturbations génétiques ou environnementales. De plus, l'interprétation des expériences est parfois délicate. Par exemple, selon les conditions de culture, l'entrée en phase stationnaire d'une culture en phase de croissance exponentielle peut être due soit à divers stress nutritionnels (manque de source de carbone, d'acides aminés, ...), soit à un excès de sous-produits toxiques du

métabolisme, ou encore à une densité de population trop élevée. Ces phénomènes sont contrôlés par des réseaux de régulation différents mais interdépendants. Nous n'avons notamment pas trouvé dans la littérature de données permettant de corroborer ou d'exclure l'existence des oscillations transitoires prédites lors d'un apport soudain de source de carbone. Nous avons toutefois identifié dans la littérature plusieurs propriétés non triviales sur le comportement du système, que nous présentons dans la suite. Toutes ces propriétés seront formulées en CTL, et testées avec NuSMV, leur vérification ou invalidation ne prenant jamais plus d'une fraction de seconde.

Le profil d'expression temporel du gène *fis*, mesuré par Northern blot, est représenté dans la figure 8.7, lorsque des cellules en phase stationnaire sont mises en présence d'une source de carbone (milieu LB) et jusqu'à leur entrée en phase stationnaire lors de l'épuisement du milieu [185]. On observe une forte augmentation de la concentration des ARNm de *fis* lors de l'entrée des cellules en phase exponentielle, puis une diminution de leur concentration, qui devient ensuite constante en fin de phase exponentielle et en phase stationnaire. D'autres travaux ont montré que la concentration cellulaire de la protéine Fis variait de façon très similaire à celle des ARNm représentée figure 8.7 [12, 15, 18]. De plus, il a été mis en évidence que ce profil ne dépendait qualitativement pas de la qualité nutritionnelle du milieu [18, 173].

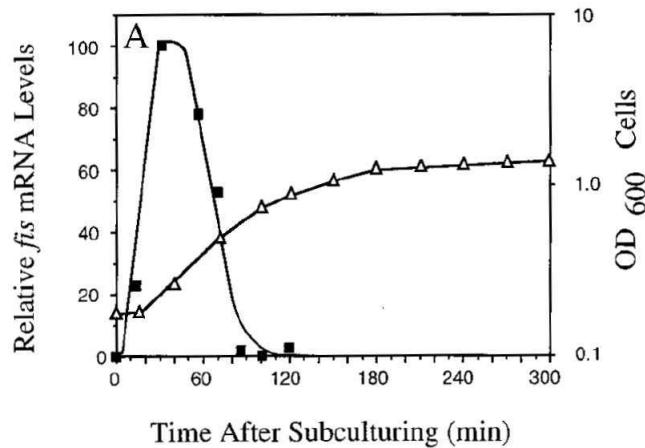


FIG. 8.7 – Profil d'expression temporel du gène *fis* consécutivement à une dilution en milieu LB d'une culture de cellules en phase stationnaire (■) [185]. La courbe de croissance (OD_{600}) est également représentée (△).

La traduction de ces propriétés en logique temporelle est quasi-immédiate, en représentant un état de la cellule en phase exponentielle ou stationnaire, par une concentration élevée ou basse des ARN stables, respectivement. Les prédictions obtenues pour la transition de la phase stationnaire à la phase exponentielle doivent satisfaire la propriété

$$\mathbf{EF}(\dot{x}_{fis} > 0 \wedge x_{rrn} > \theta_{rrn}),$$

et les prédictions obtenues pour la transition de la phase exponentielle à la phase stationnaire doivent satisfaire la propriété

$$\mathbf{EF}(\dot{x}_{fis} < 0 \wedge \mathbf{EF}(\dot{x}_{fis} = 0 \wedge x_{rrn} < \theta_{rrn})).$$

Notons que nous utilisons l'opérateur CTL \mathbf{F} , plutôt que \mathbf{G} , spécifiant qu'une propriété doit être vraie à certains moments, plutôt qu'à chaque instant, car nous ne pouvons exclure que

certaines événements prédits, notamment des oscillations transitoires de la concentration de Fis, ne soient pas observés, soit à cause d'une fréquence d'échantillonnage trop peu élevée, soit à cause de mesures faites sur des populations non synchronisées plutôt que sur des cellules individuelles. La vérification avec NuSMV de ces propriétés montre que les prédictions obtenues sont cohérentes avec ces observations.

Des informations sur la dynamique des ARN stables sont également disponibles. Fis, en régulant le promoteur P1 des opérons *rrn* contribue majoritairement à la transcription très intense des ARN stables en phase exponentielle [12, 173]. Néanmoins, on montre qu'immédiatement après un apport soudain de carbone (dilution en milieu LB), la transcription des ARN stables augmente indépendamment de Fis (voir figure 8.8). Ceci est corroboré par l'observation que la production des ARN stables augmente trop rapidement pour être le résultat d'une synthèse protéique [94].

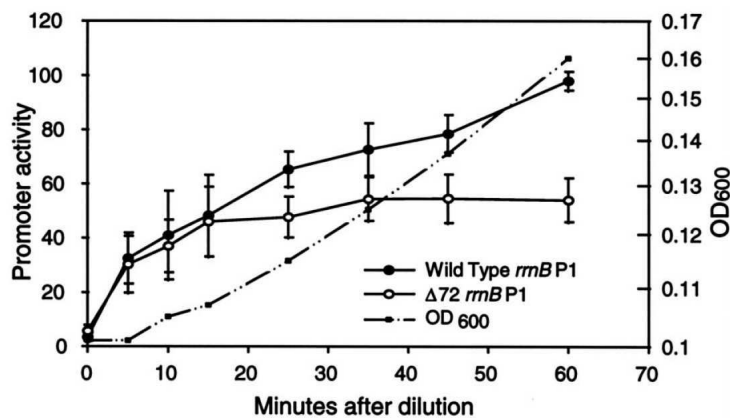


FIG. 8.8 – Activité du promoteur P1 de l'opéron *rrnB* sauvage (●) et de l'opéron *rrnB* muté dans un site de fixation de Fis (○), suite à une dilution en milieu LB d'une culture de cellules en phase stationnaire [12]. La courbe de croissance (OD₆₀₀) est également représentée.

Nous pouvons traduire cette propriété en spécifiant que l'on peut observer une augmentation de la concentration des ARN stables avant que la concentration de Fis ne soit suffisamment élevée pour activer leur synthèse. Autrement dit, il existe une exécution du système dans laquelle la concentration de Fis reste basse ($x_{fis} < \theta_{fis}^3$, où θ_{fis}^3 est la concentration au-dessus de laquelle Fis active l'expression des ARN stables) jusqu'à ce que le système atteigne un état dans lequel les ARN stables sont exprimés ($\dot{x}_{rrn} > 0$) et dans lequel $x_{fis} < \theta_{fis}^3$ reste vrai :

$$\mathbf{E}(x_{fis} < \theta_{fis}^3 \mathbf{U}(\dot{x}_{rrn} > 0 \wedge x_{fis} < \theta_{fis}^3)).$$

La vérification de cette propriété par NuSMV révèle que le modèle ne rend pas compte de cette propriété. Ainsi, d'autres régulateurs que Fis interviennent pour le contrôle de la transcription des ARN stables, et doivent être pris en compte pour expliquer ce phénomène. Dans [12], Appleman et collègues suggèrent notamment que ce phénomène peut résulter d'un contrôle par le régulateur global guanosine tetraphosphate (ppGpp). Le ppGpp est une petite molécule dont la concentration décroît très rapidement après un apport soudain de source de carbone et qui inhibe l'expression des opérons *rrn* [72, 121]. Il serait intéressant

de prendre en compte dans le modèle le ppGpp, sa synthèse et sa dégradation, pour tester si cette extension permet de résoudre le problème.

Une autre série d'expériences a montré, pour des cultures en milieu minimum avec glucose, que le niveau de superenroulement de l'ADN diminuait lors de l'entrée en phase stationnaire (voir figure 8.9) [17].

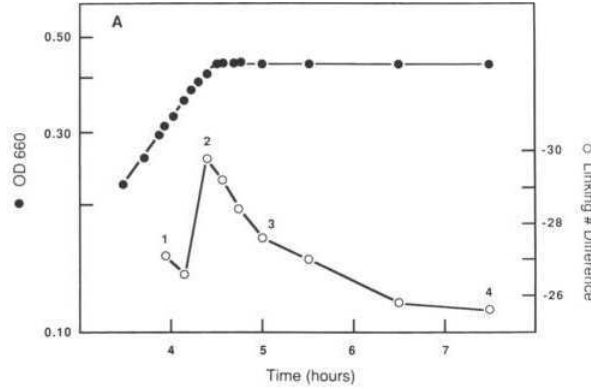


FIG. 8.9 – Evolution du niveau de superenroulement de l'ADN (○) dans une culture de cellules en milieu minimum avec glucose [17]. La courbe de croissance (OD₆₆₀) est également représentée (●).

Le niveau de superenroulement n'est pas une variable du modèle. Toutefois, l'hypothèse a été faite qu'il dépendait du ratio des concentrations de GyrAB et de TopA (section 8.1.2). Une condition nécessaire à la diminution du superenroulement est donc que, pour un instant au moins, soit la concentration de GyrAB diminue, soit la concentration de TopA augmente :

$$\mathbf{EF} (\dot{x}_{gyrAB} < 0 \vee \dot{x}_{topA} > 0).$$

Cette propriété, vérifiée avec NuSMV en une fraction de seconde, n'est pas vérifiée par les prédictions. Plus précisément on montre que l'on a $\mathbf{AG} (\dot{x}_{gyrAB} > 0 \wedge \dot{x}_{topA} = 0)$, ce qui signifie que le long de toutes les exécutions, tout le temps, le niveau de superenroulement prédit augmente ! Une extension du modèle, suggérée par la découverte de cette incohérence, est en cours de développement (Ropers *et al.*, manuscript en préparation). Ce modèle inclut, en plus des 6 gènes précédents, trois gènes codant respectivement RpoS, un autre régulateur global, impliqué dans les réponses à de nombreux stress [117], RssB, une protéine impliquée dans la dégradation de RpoS et dont l'activité dépend de l'état physiologique de la cellule [117], et GyrI, un inhibiteur de la gyrase GyrAB et dont l'expression dépend de RpoS [174]. Des résultats de simulation préliminaires montrent que ce modèle permet d'expliquer la diminution du superenroulement de l'ADN observée. C'était également cette incohérence qui nous avait conduit à corriger la concentration de GyrAB dans l'état initial (8.3) donné à la section 8.2.2.

Nous avons dans ce chapitre testé la validité d'un modèle de la réponse au stress nutritionnel, ou plus spécifiquement au manque de carbone, chez *E. coli* à l'aide de la méthode présentée dans la partie II.

Le modèle proposé par Ropers *et al.* [192] rend compte des aspects essentiels de ce phénomène, puisqu'il permet de prédire correctement les transitions entre phase exponentielle et phase stationnaire observées chez *E. coli*. Plus généralement, nous avons trouvé un

assez bon accord entre les prédictions obtenues et les données expérimentales disponibles. Toutefois, plusieurs incohérences entre prédictions et observations ont été identifiées et des extensions du modèle ont alors été proposées. Remarquons que les extensions proposées font toutes intervenir d'autres régulateurs globaux, qui sont notamment impliqués dans les réponses à d'autres stress (ppGpp, RpoS). Ceci est cohérent avec le rôle central attribué aux régulateurs globaux qui semblent coordonner le fonctionnement des réseaux de régulation contrôlant les réponses aux différents stress, et agissant souvent simultanément [117, 225].

Plus généralement, pour réellement mettre à l'épreuve la validité du modèle proposé, il est nécessaire de disposer de plus d'information sur la dynamique du système lors des transitions de la phase exponentielle vers la phase stationnaire, et inversement. Il n'a par exemple pas été possible, à partir des données disponibles dans la littérature, de corroborer ou d'infirmer la prédiction surprenante du modèle de l'existence d'oscillations amorties dans la concentration de Fis et de GyrAB lors de la réentrée des cellules en phase exponentielle. Ceci pourrait soit être dû à une fréquence d'échantillonnage trop basse, soit à des mesures sur des populations de cellules non synchrones. Ceci a motivé le développement d'un programme de *validation expérimentale* des prédictions, dont l'objectif est justement de mesurer en population et sur cellules isolées, avec une fréquence d'échantillonnage élevée et en parallèle, l'évolution des concentrations des différentes protéines du réseau.

En résumé, la méthode présentée dans la partie II et implémentée dans une nouvelle version de l'outil GNA (partie III) a été appliquée à la validation à partir de données expérimentales de deux modèles grands et complexes et pour lesquels peu de données numériques sont disponibles. Ces modèles, proposés par de Jong *et al.* [67] et par Ropers *et al.* [192], concernent respectivement l'initiation de la sporulation chez *B. subtilis* et la réponse au stress nutritionnel chez *E. coli*. Le fonctionnement global de ces deux phénomènes, bien qu'ayant été beaucoup étudiés, n'est encore que partiellement compris. Nous avons pu vérifier par model checking que, dans les deux cas, les prédictions obtenues étaient globalement cohérentes avec la plupart des propriétés observées du système. Toutefois, plusieurs incohérences ont également été identifiées, suggérant des révisions ou des extensions des modèles, ainsi que la réalisation d'expériences complémentaires. Egalement, des prédictions surprenantes ont été obtenues pour la réponse au stress nutritionnel chez *E. coli*, qui ne peuvent être ni corroborées ni invalidées à partir des données expérimentales disponibles dans la littérature, ce qui a motivé la mise en place d'un programme de validation expérimentale des prédictions.

En dehors d'une contribution à une meilleure compréhension du fonctionnement de ces systèmes, ces deux études de cas illustrent plus généralement l'applicabilité à l'analyse de système d'intérêt biologique de l'approche que nous proposons, utilisant une méthode et une implémentation dédiées. Premièrement, il est ainsi possible d'obtenir des prédictions sur le comportement de systèmes biologiques complexes. Malgré le nombre relativement élevé de variables des modèles analysés, les temps de calcul des prédictions restent de l'ordre de la dizaine de secondes. Deuxièmement, il est possible de formaliser en logique temporelle un large éventail de propriétés observées sur ces systèmes. De nombreuses propriétés biologiquement intéressantes font notamment intervenir les sens de variation des concentrations des protéines. Enfin, nous avons montré qu'il est possible de tester efficacement la cohérence entre les prédictions obtenues et les données expérimentales, les temps de calcul pour la vérification des propriétés étant de l'ordre de la seconde.

Cinquième partie

Discussion et conclusions

9 Discussion

Nous avons présenté une méthode pour la validation de modèles LPM de réseaux de régulation génique, son implémentation et son application à la validation de deux modèles grands et complexes. Cette discussion a pour objectif de replacer le travail présenté précédemment dans le contexte des travaux similaires et de discuter les avantages et inconvénients relatifs des différentes approches.

Nous commençons par résumer brièvement les principales caractéristiques de notre approche. L'approche présentée repose sur trois points essentiels. Premièrement, l'utilisation d'*abstraction discrète* permet d'établir une connexion à la fois intuitive et mathématiquement bien fondée entre la définition continue de la dynamique du système, donnée en termes d'équations différentielles LPM, et sa représentation discrète, donnée par le graphe de transitions entre états. Deuxièmement, l'utilisation de *model checking* permet de formaliser un large éventail de propriétés dynamiques et de les vérifier efficacement. Enfin, les applications biologiques, en plus d'apporter des résultats biologiquement intéressants, montrent plus généralement l'*applicabilité* de l'approche aux systèmes d'intérêt biologique.

En comparaison des approches proposées dans le domaine, très vaste, de l'analyse de modèles de réseaux biologiques, notre approche présente une double originalité. Premièrement, peu d'approches se sont intéressées spécifiquement au problème, pourtant important, de la validation de modèle. Deuxièmement, aucune autre approche, à notre connaissance, ne semble avoir utilisé conjointement l'abstraction discrète et le model checking pour l'analyse de réseaux biologiques.

Par contre, un grand nombre de travaux ont soit utilisé l'abstraction discrète pour analyser des modèles continus, soit le model checking pour vérifier les propriétés dynamiques de modèles discrets. Nous organisons donc notre discussion selon ces deux axes. Etant donné le nombre et la diversité des approches concernées, nous avons choisi de structurer notre discussion autour de la présentation détaillée d'un nombre restreint de travaux, permettant d'illustrer clairement les points forts et faibles des différentes approches.

9.1 Analyse de modèles continus par abstraction discrète

9.1.1 Modèles continus

Les modèles continus définis en termes d'équations différentielles non linéaires sont très fréquemment utilisés pour représenter les réseaux biologiques. Les modèles étudiés sont en général numériques, car leur analyse analytique est notoirement difficile. Un problème bien connu de cette approche est qu'il est pratiquement impossible d'invalider ces modèles ayant un grand nombre de paramètres inconnus, car il est alors presque toujours possible de trouver un jeu de paramètres tel que les prédictions correspondent aux données [227], cette correspondance étant en général établie visuellement par « superposition » des courbes. Et quand bien même un tel jeu de paramètres n'est pas trouvé, cela n'invalide pas pour autant le modèle, car tous les jeux de paramètres et toutes les conditions initiales possibles ne peuvent être testés. Lorsque ces méthodes sont appliquées à des modèles complexes et à des systèmes pour lesquels peu d'information quantitative est disponible, la pertinence biologique des résultats obtenus est alors sujette à caution.

Une solution à ce problème est le développement d’approches qui permettent d’obtenir des prédictions valables pour un ensemble de conditions initiales et/ou pour un ensemble de valeurs de paramètres. Diverses méthodes numériques ont été proposées, notamment par Belta *et al.* [29], qui utilisent des techniques de systèmes hybrides pour le calcul d’atteignabilité,¹ par El-Samad *et al.* [77], qui utilisent des techniques d’optimisation appelées SOSTOOLS, et par Eveillard *et al.* [84], qui utilisent de la programmation par contraintes hybride. Dans ces approches, l’utilisation d’intervalles pour spécifier les conditions initiales ou les valeurs de paramètres permet de traduire les incertitudes sur leurs valeurs exactes. En plus de ces méthodes numériques, l’utilisation de méthodes symboliques ou qualitatives a été proposée. L’avantage de ces dernières approches est naturellement de s’affranchir plus encore de la nécessité de devoir fournir des valeurs pour les conditions initiales ou pour les paramètres. Ces méthodes reposent sur la notion d’abstraction discrète.

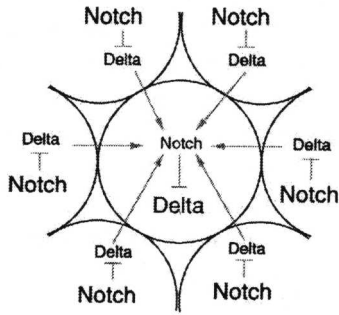
9.1.2 Utilisation d’abstraction discrète

Les techniques d’abstractions discrètes ont été utilisées par plusieurs groupes de chercheurs pour l’analyse de réseaux biologiques dans les communautés du raisonnement qualitatif et des systèmes hybrides. Il est à noter que des liens entre ces deux théories ont été par ailleurs établis [202, 139]. Nous décrivons plus particulièrement une méthode proposée par Ghosh et Tomlin [97] pour l’analyse symbolique de modèles LPM représentés sous forme d’automates hybrides affines par morceaux, et qui est adaptée à l’analyse de réseaux biologiques. Cette méthode a été appliquée à l’analyse d’un phénomène d’inhibition latérale médié par deux protéines, Delta et Notch, et modélisé comme un réseau de régulation génique, suivant Marnellos *et al.* [155] (voir figure 9.1).

Dans cette approche, l’espace d’état continu est partitionné en régions par un ensemble de polynômes, définissant les frontières de ces régions et une méthode est proposée pour calculer, à partir de cette partition, un système de transitions quotient. Cette méthode utilise pour le calcul des transitions des raisonnements qualitatifs et une procédure de décision permettant de calculer les dérivées de Lie des polynômes définissant la frontière des régions (voir figures 9.2(a) et (b)). Cette procédure de décision est supportée par l’outil d’élimination des quantificateurs QEPCAD. Lorsqu’un état discret possède plus d’un successeur, la région correspondante peut, dans certains cas, être repartitionnée de telle sorte que les sous-régions soient chacune associée à un état discret ayant au plus un successeur (voir figure 9.2(c)). Cette procédure nécessite le calcul de solutions analytiques des équations différentielles, ce qui n’est pas toujours réalisable en pratique.

Un modèle hybride pour le phénomène d’inhibition latérale étudié, représentant les interactions entre 4 cellules, et ayant 8 variables continues et initialement 256 états discrets, a été analysé par cette méthode. Dans un premier temps, les états d’équilibres « biologiquement réalisables » (*i.e.* réguliers dans la terminologie que nous utilisons) du système ont été identifiés. Puis, pour chacun de ces états d’équilibres, est calculé l’ensemble des états ayant un successeur unique et à partir desquels il est possible d’atteindre l’état d’équilibre considéré. On obtient ainsi une sous-approximation de l’ensemble des états initiaux pour lesquels le système atteint nécessairement l’état d’équilibre (« uniquely backward reachable states »). Les résultats obtenus, existence sous certaines conditions d’une amplification des différences entre cellules, sont en accord avec les données biologiques disponibles et avec

¹D’autres applications biologiques de la théorie des systèmes hybrides ont été proposées dans [11, 14, 28, 126]. Il est à noter qu’Antonioti *et al.* [11] utilisent également des techniques de model checking.



(a)

$$\begin{aligned}
 & H_{\text{one_cell}} = (Q, X, \Sigma, \text{Init}, f, \text{Inv}, R) \text{ où :} \\
 & - Q = q_1, q_2, q_3, q_4, \\
 & - X = (x_1, x_2)^T \in \mathbb{R}^2, \\
 & - \Sigma = \{u_N = \sum_{i=1}^6 x_{\text{Delta},i}\}, \\
 & - \text{Init} = Q \times \{X \subset \mathbb{R}^2 \mid x_1, x_2 > 0\}, \\
 & - f(q, x) = \begin{cases} [-\lambda_D x_1, -\lambda_N x_2]^T, & \text{si } q = q_1 \\ [R_D - \lambda_D x_1, -\lambda_N x_2]^T, & \text{si } q = q_2 \\ [-\lambda_D x_1, R_N - \lambda_N x_2]^T, & \text{si } q = q_3 \\ [R_D - \lambda_D x_1, R_N - \lambda_N x_2]^T, & \text{si } q = q_4 \end{cases} \\
 & - \text{Inv} = \{q_1, \{-x_2 < h_D, u_N < h_N\}\} \cup \\
 & \quad \{q_2, \{-x_2 \geq h_D, u_N < h_N\}\} \cup \\
 & \quad \{q_3, \{-x_2 < h_D, u_N \geq h_N\}\} \cup \\
 & \quad \{q_4, \{-x_2 \geq h_D, u_N \geq h_N\}\}, \\
 & - R : \left[\begin{array}{l} R(q_1, \{u_D \geq h_D \wedge u_N < h_N\}) \in q_2 \times \mathbb{R}^2 \\ R(q_1, \{u_D < h_D \wedge u_N \geq h_N\}) \in q_3 \times \mathbb{R}^2 \\ R(q_1, \{u_D \geq h_D \wedge u_N \geq h_N\}) \in q_4 \times \mathbb{R}^2 \\ R(q_2, \{u_D \geq h_D \wedge u_N \geq h_N\}) \in q_4 \times \mathbb{R}^2 \\ R(q_3, \{u_D < h_D \wedge u_N < h_N\}) \in q_1 \times \mathbb{R}^2 \\ R(q_4, \{u_D < h_D \wedge u_N < h_N\}) \in q_1 \times \mathbb{R}^2 \\ R(q_4, \{u_D \geq h_D \wedge u_N < h_N\}) \in q_2 \times \mathbb{R}^2 \\ R(q_4, \{u_D < h_D \wedge u_N \geq h_N\}) \in q_3 \times \mathbb{R}^2 \end{array} \right] \\
 & h_D, h_N : -R_N/\lambda_N < h_N \leq 0 \wedge 0 < h_N \leq R_D/\lambda_D
 \end{aligned}$$

(b)

FIG. 9.1 – (a) Réseau intercellulaire de signalisation entre les protéines Delta et Notch, pour une cellule et ses 6 voisins. (b) Automate hybride $H_{\text{one_cell}}$ modélisant les interactions entre les protéines Delta et Notch dans une cellule. [97].

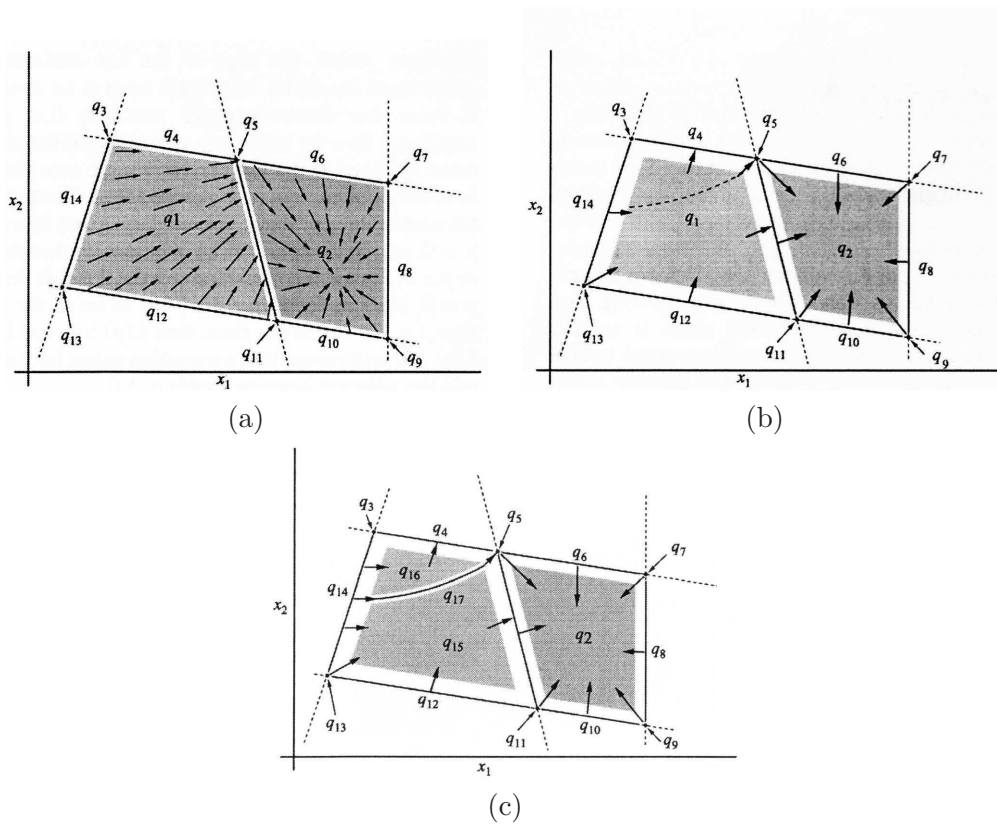


FIG. 9.2 – (a) Représentation du flot dans l’espace d’état continu, et de la partition initiale d’un automate hybride de dimension 2. (b) Représentation des transitions calculées entre états discrets. (c) Partition de l’état q_1 , ayant plus d’un successeur, en trois états, q_{15} , q_{16} et q_{17} , ayant chacun un unique successeur. [97].

les résultats d'autres études [62, 155].

La notion d'abstraction discrète, également appelée abstraction qualitative, est aussi utilisée dans la *simulation qualitative*. Cette méthode, développée par Kuipers et collègues, permet d'obtenir des prédictions sur la dynamique du système en l'absence d'information quantitative, en déduisant l'ensemble des comportements qualitatifs possibles, de la structure du système [66, 141, 218]. Dans cette approche, les systèmes sont décrits par des équations différentielles qualitatives. La simulation qualitative, ou des approches similaires, ont été utilisées pour l'analyse de différents systèmes biologiques par Heidtke et Schulze-Kremer [115], par King *et al.* [134] en combinaison avec des techniques d'apprentissage, par Kim et Tidor [133] pour faire de la validation de modèles de réseaux de régulation génétique simples en comparant les changements, prédits et observés, de l'état d'équilibre de réseaux soumis à des perturbations, et par Siegel *et al.* [198].

Enfin, bien que le domaine d'application biologique (écologie) et le type de modèles considérés (« diagonally monotonic loop-structured systems ») soient très différents, une méthode qualitative développée par Bernard et collègues [31, 32] présente de nombreuses similarités avec notre approche. Effectivement, dans [31], Bernard et Gouzé proposent une méthode qualitative pour calculer une partition de l'espace des phases, qui préserve entre autres, les signes des dérivées, et pour calculer les transitions entre régions. Un graphe de transitions entre états discrets est également obtenu par abstraction discrète. De plus, dans [32], Bernard et Souissi appliquent cette méthode à la validation de modèles écologiques en comparant la succession des extremas (maximas et minimas) prédits et observés. Une telle comparaison entre les prédictions obtenues et les données biologiques, pourtant révélatrice de la qualité des prédictions, est rarement réalisée.

9.1.3 Comparaison et évaluation

Comme nous, Ghosh et Tomlin utilisent des modèles LPM. Les différences entre les deux approches portent sur les méthodes d'analyse employées. Ces auteurs utilisent des méthodes plus puissantes, mais aussi plus coûteuses en calculs. Il est ainsi par exemple possible d'exprimer et d'exploiter le fait que deux paramètres de synthèse ou de dégradation soient égaux (par exemple, pour le même gène dans deux cellules différentes). Également, il est possible de considérer des partitions non hyperrectangulaires et des raffinements successifs de ces partitions. Toutefois, ces calculs reposent sur l'utilisation de procédures de décision complexes. Il est donc fort probable que le prix à payer de ce gain en généralité de la méthode d'analyse soit une moins bonne mise à l'échelle. Il est également important de remarquer que la partition proposée par Ghosh et Tomlin [97] et celle que nous proposons dans ce travail sont deux raffinements différents de la partition utilisée par de Jong *et al.* [70] (*i.e.* partition en domaines de modes), l'une visant à obtenir des propriétés d'atteignabilité unique, et l'autre, des propriétés de monotonie unique. Il ne semble toutefois pas y avoir de problème théorique à l'obtention d'une partition combinant les deux raffinements à l'aide de la méthode proposée par Ghosh et Tomlin [97].

Dans des travaux préliminaires sur un modèle semblable à celui décrit plus haut, mais n'ayant que 4 variables continues, un état qualifié d'état *Zeno* a été détecté à l'intersection de plans de seuils, où le champ de vecteurs définissant la dynamique du système présente des discontinuités [96]. Le problème rencontré est le même que celui que nous avons rencontré au chapitre 2 lors de la définition de l'automate hybride affine par morceau correspondant au système à deux réservoirs qui nous a servi d'exemple dans ce chapitre. Nous avons vu qu'en présence de certaines discontinuités du champ de vecteurs, il n'est

pas toujours évident de définir les solutions du système (section 2.1.2).

Pour traiter ces problèmes, nous avons proposé au chapitre 3, d'utiliser des inclusions différentielles pour définir la dynamique du système au niveau des plans de seuils. Nous avons montré sur l'exemple du réseau d'inhibition réciproque, qui possède une dynamique très similaire au système à deux réservoirs, comment cette méthode permettait effectivement de résoudre ce problème. Si on analyse avec GNA le modèle à 4 variables continues proposée par Ghosh et Tomlin dans [96], l'état qualifié précédemment de Zeno est détecté comme un état d'équilibre singulier. On montre de plus par des critères simples, que cet état d'équilibre est instable [44]. Cette caractérisation représente bien les phénomènes continus ayant lieu, puisque dans un modèle non linéaire de ce système, proposé par Collier *et al.* [62], cet état correspond à un point selle.

En plus des équilibres singuliers, il existe d'autres phénomènes dynamiques importants pouvant avoir lieu au niveau de seuils de concentration, notamment les modes glissants, qui ne sont pas pris en compte par la méthode présentée dans [97]. Dans le contexte particulier de la validation de modèle, ceci a pour conséquence que des phénomènes observés pourraient ne pas trouver de correspondance dans les prédictions. Ainsi, des modèles risquent être rejetés à tort. Récemment, Adélaïde et Sutre [1] ont proposé une approche similaire à celle de Ghosh et Tomlin, mais qui, comme nous, utilisent des inclusions différentielles pour prendre en compte les phénomènes ayant lieu au niveau des seuils. L'analyse proposée ne s'applique toutefois qu'à une classe restreinte de modèles. Enfin, l'applicabilité de cette méthode à l'analyse de modèles grands et complexes n'a pas encore été démontrée.

Nous avons vu dans le paragraphe précédent que des approches de simulation qualitative, utilisant également des abstractions discrètes, ont été appliquées à l'analyse de systèmes biologiques. En comparaison avec notre travail, ces approches utilisent une classe de modèles beaucoup plus générale que les modèles LPM que nous utilisons. Un avantage est qu'il est ainsi possible de représenter et d'analyser une très grande variété de systèmes biologiques. Un inconvénient est que les informations disponibles n'offrent en général que des contraintes faibles sur la dynamique du système. En conséquence, la présence de nombreux états, transitions et comportements artéfactuels, et l'explosion du nombre d'états atteignables limitent habituellement l'utilisation de la simulation qualitative [66, 193].

Bien que les modèles continus permettent de représenter naturellement les phénomènes biologiques et sont parmi les plus utilisés, le manque de données quantitative d'une part, et la difficulté de leur analyse analytique d'autre part, limitent leur application. Dans ce contexte, l'utilisation d'abstraction discrète est particulièrement intéressante pour l'analyse d'une certaine classe de modèles, dont les modèles LPM, puisqu'elle permet d'obtenir par des calculs symboliques ou qualitatifs une représentation discrète de la dynamique du système, qui peut alors être analysée par des techniques informatiques classiques.

Il ressort également de cette discussion que le choix du niveau d'abstraction est crucial. Trop élevé, les prédictions obtenues peuvent être excessivement conservatives, donc peu significatives, et peu discriminantes dans le contexte de la validation de modèle. Trop fin, le calcul de la représentation discrète peut devenir prohibitivement complexe, limitant l'application de la méthode à des modèles simples.

Une autre direction possible consiste à utiliser directement des modèles discrets, plutôt que des modèles continus en combinaison avec de l'abstraction discrète. Nous discutons dans la section suivante de ces approches, de leurs avantages, notamment la simplicité de

l'analyse, mais aussi de leurs inconvénients.

9.2 Analyse de modèles discrets par model checking

9.2.1 Modèles discrets

En réponse à la difficulté de l'analyse des modèles continus, de nombreux formalismes discrets ont été proposés pour la modélisation et l'analyse de la dynamique des réseaux biologiques (voir [65] pour une revue). Parmi ces formalismes figurent notamment les *approches logiques* dont les réseaux booléens [131, 209] et l'approche logique généralisée [212, 214], les *approches basées sur la notion de concurrence* dont les algèbres de processus [95], les systèmes de transitions concurrents [46] et les logiques de réécriture [76], et les approches utilisant les *réseaux de Petri* [47, 116, 189, 228].²

Toutes les approches mentionnées ci-dessus ont en commun de représenter l'état des différents éléments du réseau par une variable discrète. Ainsi, un gène est considéré comme soit actif, soit inactif, et une espèce protéique, comme soit présente, soit absente. L'état du système est alors donné par un vecteur représentant la valeur de toutes les variables.

Les approches mentionnées ci-dessus diffèrent sur plusieurs aspects. Tout d'abord, les hypothèses biologiques utilisées dépendent du type de réseau considéré. Egalement, ces approches diffèrent sur la façon dont est représentée la logique des phénomènes biologiques étudiés (régulation génique, interaction entre protéines, . . .) et dont est définie l'évolution discrète du système (*i.e.* les transitions entre deux états discrets). Enfin, les possibilités de simulation et d'analyse offertes varient significativement selon le formalisme utilisé. Mais dans tous les cas, la dynamique du système est, ou peut être, représentée par un graphe de transitions entre états discrets, ce qui rend applicables les techniques de model checking. Les possibilités offertes par ces techniques pour l'analyse efficace de grands modèles ont motivés divers groupes de chercheurs à étendre des approches discrètes dans cette direction.

9.2.2 Utilisation du model checking

Trois groupes de chercheurs ont effectivement utilisé le model checking, soit pour analyser des modèles comprenant un grand nombre de variables (Chabrier-Rivier *et al.* [46] et Eker *et al.* [76]), soit pour analyser systématiquement un grand nombre de modèles simples (Bernot *et al.* [33]). Dans la suite, nous discutons plus spécifiquement des travaux de Bernot et collègues, qui présentent de nombreux points communs avec l'approche que nous avons proposée. Comme nous le verrons, les différences entre les deux approches sont subtiles mais importantes. Ces auteurs utilisent l'approche logique généralisée, ou plus exactement une reformulation plus adaptée à un traitement informatique de celle-ci, en combinaison avec du model checking, pour faire de la sélection de modèles [33]. Nous commençons par décrire très succinctement l'approche logique généralisée (voir [213, 214, 215] pour une description plus détaillée).

Les approches logiques ont été spécifiquement développées pour l'analyse de la dynamique de réseaux de régulation génique. Dans l'approche logique généralisée proposée par Thomas et collègues [212, 214], des variables discrètes, dites logiques ou multivaluées, abstraient les concentrations réelles des protéines. Le système est modélisé par un graphe de

²Des approches de programmation par contraintes, inspirées de l'approche logique généralisée, ont également été récemment proposées [61, 86]

régulation biologique G , qui est un graphe orienté et étiqueté représentant les interactions entre gènes, et par un ensemble de paramètres donnant pour chaque variable v et pour chaque état du système (ou plutôt pour chaque état de l'ensemble des prédécesseurs de v dans le graphe G), l'image de v , c'est-à-dire la valeur vers laquelle tend v (figure 9.3(a)³). Bien qu'ils soient formulés différemment, ces modèles et nos modèles qualitatifs, constitués d'équations différentielles LPM et de contraintes d'inégalité sur les paramètres, sont basés sur des approximations biologiques similaires.

A un réseau de régulation biologique, on associe un graphe d'état synchrone dont les sommets sont les états du réseau et les arcs associent chaque état à son successeur (figure 9.3(b)). Un graphe d'état asynchrone est alors obtenu en "désynchronisant" les transitions. La désynchronisation est une opération qui garantit que lors d'une transition (1) une seule variable logique change de valeur et (2) la valeur d'une variable ne peut changer au plus que d'une unité (figure 9.3(c)). C'est ce graphe asynchrone qui définit la dynamique du système.

Il est important de remarquer ici que l'approche logique généralisée a été développée en référence à des modèles continus. Les règles de calcul du graphe de transition ont été élaborées de façon à ce que le graphe de transitions asynchrone obtenu capture intuitivement la dynamique des modèles continus sous-jacents. Des liens explicites entre l'approche logique généralisée et l'approche par équations différentielles LPM ont également été établis par Snoussi. En particulier, ce dernier a montré que la méthode logique généralisée peut être considérée comme résultant d'une discrétisation de systèmes continus à équations différentielles LPM [200]. Dans la terminologie que nous utilisons, les états logiques correspondent aux domaines de mode *réguliers*, et les graphes asynchrones sont analogues, mais non identiques, aux graphes de transitions obtenus par la méthode proposée par de Jong *et al.* [70]. En particulier, ces graphes n'incluent pas les domaines de mode singuliers.

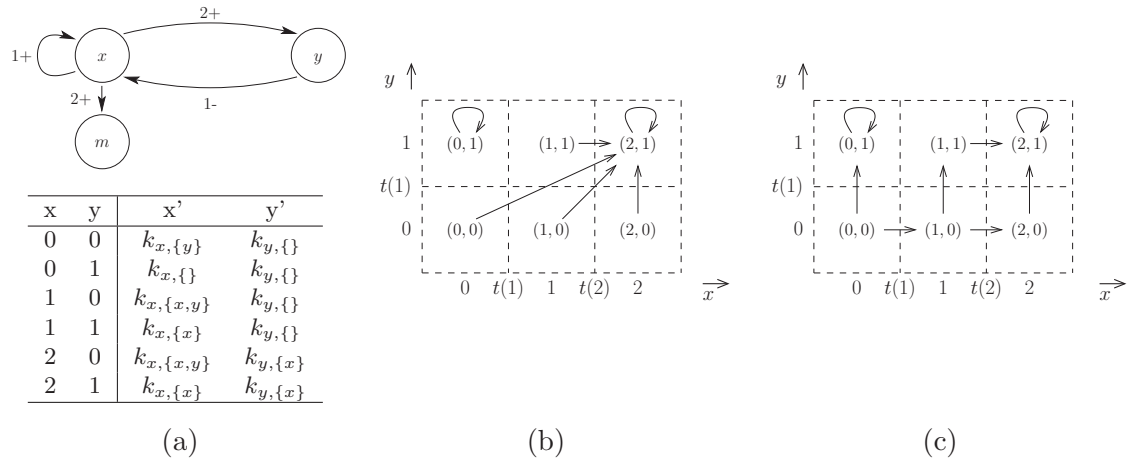


FIG. 9.3 – (a) Réseau de régulation biologique constitué d'un graphe de régulation biologique (m représente le mucus) et d'un ensemble de paramètres (l'attracteur de l'état (x,y) est (x',y')). Les valeurs discrètes des paramètres utilisées sont $k_{x,\{\}} = 0$; $k_{x,\{x\}} = 2$; $k_{x,\{y\}} = 2$; $k_{x,\{x,y\}} = 2$; $k_{y,\{\}} = 1$; $k_{y,\{x\}} = 1$. (b) Graphe de transition asynchrone. (c) Graphe de transitions synchrone obtenu à partir de (b). [33].

³Les paramètres $k_{i,\dots}$ dans la figure 9.3 sont analogues dans les modèles LPM aux concentrations focales, ψ_i , et ne doivent pas être confondus avec les taux de synthèse, κ_i .

L'approche présentée par Bernot *et al.* [33] permet de sélectionner parmi un ensemble de modèles potentiels les modèles satisfaisant un ensemble de contraintes sur les paramètres traduisant diverses hypothèses biologiques et de modélisation, et un ensemble de propriétés dynamiques formulées en CTL et exprimant des contraintes sur les valeurs discrètes des variables. L'approche est implémentée dans le logiciel SMBioNet, utilisant NuSMV. La méthode a été appliquée à l'analyse d'un ensemble de 648 réseaux potentiels simples, tous composés de deux gènes impliqués dans la production de mucus chez *P. aeruginosa* (voir figure 9.3(a) pour un exemple). Ces modèles ont ensuite été testés selon leur capacité à soutenir l'hypothèse biologique que la mucoïdie peut être induite par un phénomène épigénétique. L'analyse réalisée a montré que certains modèles potentiels satisfont toutes les contraintes données, et que donc sur la base des connaissances disponibles, cette hypothèse ne pouvait être écartée. Une validation expérimentale est en cours.

Dans un contexte biologique différent, l'étude de réseaux d'interaction protéine/protéine et de réseaux de transduction de signaux, Chabrier-Rivier *et al.* [46] et Eker *et al.* [76] ont également utilisé le model checking pour l'analyse de modèles discrets. Dans [46], Chabrier-Rivier et collègues proposent un formalisme simple et lisible essentiellement dédié à la modélisation de réseaux d'interaction protéique. Ce formalisme peut être traduit en système de transitions concurrent pour permettre son analyse par model-checking. Un modèle du cycle cellulaire faisant intervenir un grand nombre d'espèces moléculaires et de règles de réactions est proposé et diverses propriétés dynamiques du système, exprimées en CTL, sont analysées grâce à l'outil BioCham [85], utilisant NuSMV. Pour l'exemple proposé, une sémantique booléenne (*i.e.* présence/absence) est utilisée. Récemment, cette approche a été utilisée par Calzone *et al.* [43] en combinaison avec des techniques d'apprentissage automatique pour la révision de modèles. Des applications sur des modèles du cycle cellulaire montrent l'intérêt biologique potentiel de cette méthode. Une approche similaire à [46] est proposée par Eker et collègues [76]. Toutefois, ces derniers utilisent un langage très général (logiques de réécritures) pour la modélisation et une boîte à outils, contenant un model checker, non dédiée à l'analyse des systèmes biologiques (Maude) pour l'analyse.

Enfin, mentionnons les travaux de Heiner, Koch et collègues [116, 136] portant spécifiquement sur la validation de modèles de réseaux biologiques, représentés sous forme de réseaux de Petri discrets. Les réseaux de Petri possèdent à la fois une sémantique précisément définie et une représentation graphique intuitive. Dans cette approche, des techniques spécifiques à l'analyse des réseaux de Petri ont été utilisées. Par la méthode de validation proposée, il est essentiellement possible de vérifier la *cohérence interne* du modèle, notamment des propriétés de structure et d'invariants. En particulier, il n'est pas possible de vérifier le type de propriétés exprimables en logique temporelle, portant sur la dynamique du système.

9.2.3 Comparaison et évaluation

Les principales différences entre les approches discrètes, notamment l'approche logique généralisée, et l'approche que nous proposons, peuvent être facilement illustrées en comparant les différents graphes de transitions entre états discrets, c'est-à-dire les descriptions de la dynamique du système que l'on obtient par les différentes approches. Nous distinguons trois points.

Une première différence est la sémantique des graphes de transitions. Dans l'approche que nous proposons, comme dans celle proposée précédemment par de Jong *et al.* [70], une transition entre deux états discrets est définie par l'existence d'une solution du sys-

tème continu passant par les deux domaines correspondants aux états. Cette définition découle directement de l'utilisation de l'abstraction discrète (voir la définition du système de transitions quotient, section 4.2.2). Dans les approches discrètes, une transition entre deux états discrets est définie par des règles de transition, motivée par l'intuition des phénomènes continus sous-jacents. Remarquons, que dans toutes les approches présentées, le calcul effectif du graphe se fait par l'intermédiaire de règles de transitions. Toutefois, dans notre approche, les règles utilisées ont été démontrées à partir de la définition du système continu. L'utilisation d'abstraction discrète permet ainsi d'établir clairement le lien entre la dynamique continue et sa représentation discrète.

Une seconde différence est que la dynamique du système au niveau des seuils de concentrations, qui sont situés pour les approches discrètes « entre » deux valeurs discrètes, n'est pas prise en compte dans les travaux mentionnés ci-dessus. Ceci se traduit, dans l'approche logique généralisée par l'absence des domaines de mode singuliers dans le graphe de transitions. Nous rappelons que dans le cadre de la validation de modèle, la prise en compte des phénomènes dynamiques pouvant avoir lieu au niveau des seuils est essentielle pour garantir qu'un modèle ne peut être rejeté à tort (section 9.1.3). Dans le cadre de l'approche logique généralisée, une solution partielle à ce problème a été proposée par Snoussi et Thomas pour la détection des états d'équilibre singuliers [201]. Egalement, des extensions, non utilisées par Bernot et collègues dans [33], ont été récemment proposées par Richard *et al.* [191] via l'introduction d'*états singuliers* pour représenter la dynamique du système au niveau des seuils. Notons que les approches présentées par Richard *et al.* [191] et par de Jong *et al.* [70], bien que basées toutes les deux sur les mêmes systèmes d'équations différentielles LPM et sur un même niveau d'abstraction, n'aboutissent pas à des définitions de transition entre états équivalentes.

Plus généralement, le fait de représenter explicitement le système au niveau continu, puis d'utiliser une abstraction discrète, permet de traiter les problèmes rencontrés dans la représentation la plus adaptée. Ainsi, le problème de la définition de la dynamique du système au niveau des seuils de concentration est traité au niveau continu à l'aide d'inclusions différentielles, tandis que le problème de la vérification des propriétés dynamiques est traité au niveau discret par model checking.

Une troisième différence est que les niveaux d'abstraction proposés dans les différentes approches ne sont pas identiques. Les prédictions obtenues avec les méthodes présentées par Chabrier-Rivier *et al.* [46] et par Eker *et al.* [76] sont de nature booléennes (présence/absence), et celles obtenues avec la méthode présentée par Bernot *et al.* [33] sont d'une finesse similaire à celle obtenue en utilisant les domaines de mode (section 9.2.2). La méthode que nous présentons, qui est de granularité plus fine puisqu'elle utilise des domaines de flot qui sont un raffinement des domaines de mode, donne donc des résultats plus précis que les approches précédentes. Le prix à payer pour ce gain de précision est une augmentation du nombre des états, qui reste en pratique modérée (voir la section 7.2.2). Ce raffinement permet de pouvoir prédire les sens de variation des variables, et permet ainsi de comparer les prédictions obtenues avec les données expérimentales disponibles. Ceci est essentiel dans le contexte de la validation de modèles. Nous rappelons ici que le niveau d'abstraction utilisé dans les approches proposées par Bernot *et al.* [33] et par de Jong *et al.* [70] n'est pas suffisamment fin pour pouvoir prédire les sens de variation des variables. Il est facile d'illustrer cette idée en considérant l'état discret $(1, 1)$, qui a pour attracteur l'état $(2, 1)$ (figure 9.3(b)). La variable logique y est constante (*i.e.* $y' = y$ dans la figure 9.3(a)). Pourtant, les solutions d'un modèle LPM correspondant au modèle logique

proposé ne sont pas toutes constantes dans la dimension de y . On peut même montrer que les solutions qui partent de l'état (1.0) sont toutes strictement croissantes dans cette dimension (comme suggéré dans la figure 9.3(b)). Ce résultat, qui utilise des propriétés de monotonie et de convergence asymptotique des solutions, est directement obtenu par notre méthode, et semble difficile à justifier dans un formalisme purement discret.

En dehors de ces limitations, des perspectives intéressantes sont offertes par les travaux de Bernot *et al.* [33], sur la sélection de modèles, et de Calzone *et al.* [43], sur l'inférence de modèles. Ces problèmes sont complémentaires avec le problème de la validation de modèle que nous avons considéré dans ce travail. Le développement de méthodes efficaces pour faire de la sélection, et plus encore, de l'inférence de modèle, est un problème très intéressant, tant du point de vue informatique que biologique.

En résumé, la discussion présentée dans ce chapitre fait apparaître deux caractéristiques originales de notre approche.

Premièrement, nous utilisons une abstraction discrète en combinaison avec du model checking. L'utilisation de ces deux techniques offre des avantages complémentaires, qui ensemble, permettent l'analyse des propriétés dynamiques (d'une certaine classe) de grands modèles continus. Plus précisément, l'abstraction discrète permet de donner une signification intuitive et une sémantique mathématiquement bien fondée au graphe de transitions décrivant la dynamique du système, et le model checking permet l'analyse efficace des grands graphes obtenus par abstraction discrète.

Deuxièmement, nous utilisons un niveau d'abstraction relativement fin qui permet d'obtenir une description des comportements du système adaptée à la comparaison avec les données expérimentales disponibles (prédiction des sens de variation des concentrations des protéines), mais qui ne nécessite toutefois pas l'utilisation de procédures de décision complexes pour le calcul des prédictions.

10 Conclusions

Dans ce dernier chapitre, nous faisons le bilan des travaux réalisés, puis nous indiquons plusieurs directions de travail futur.

10.1 Bilan

Dans ce travail, nous avons proposé une méthode pour la validation de modèles de réseaux de régulation génique. La compréhension du fonctionnement de ces réseaux est un problème biologique important, car ils contrôlent le développement et le fonctionnement des organismes vivants. Etant donné que les réseaux d'intérêt biologique impliquent souvent un grand nombre de gènes qui interagissent par des boucles de rétroaction complexes, il est en général difficile de comprendre par des raisonnements intuitifs comment la dynamique du système émerge des interactions entre ses différents composants. L'utilisation de modèles est alors nécessaire.

Partant de ce constat, de nombreuses méthodes ont été proposées pour la modélisation et la simulation des réseaux de régulation génique. Etonnamment, le problème de la validation des modèles n'a attiré jusqu'à récemment que peu d'attention. Pourtant, cette étape est d'autant plus importante que les réseaux de régulation génique modélisés sont complexes et encore imparfaitement connus.

On dit qu'un modèle est *valide* si on ne peut déduire du modèle une contradiction avec l'ensemble des connaissances sur le système. En particulier, les prédictions obtenues à partir du modèle doivent être en accord avec les observations faites sur le système. Etant donné que les réseaux de régulation génique sont grands, que leur dynamique est complexe, et que les données expérimentales disponibles sont en général peu fiables, tester la validité d'un modèle de réseau de régulation génique est une tâche difficile, et sujette à l'erreur. Dans ces conditions, réaliser ce test de façon intuitive et informelle peut entraîner le rejet à tort d'un modèle. Ce sont ces raisons qui ont motivé le développement de la méthode proposée dans ce travail.

Nous avons considéré dans ce travail des modèles qualitatifs et hybrides, définis par des équations différentielles LPM. En exploitant la forme mathématique simple de ces modèles et des informations qualitatives sur les valeurs des paramètres qui peuvent habituellement être déduites de la littérature expérimentale, il est possible de calculer symboliquement une représentation discrète de la dynamique du système dans l'espace des phases, sous la forme d'un graphe de transitions [22, 69, 70]. Plus précisément, le graphe de transitions est obtenu par abstraction discrète à partir d'une partition de l'espace des phases en régions appelées domaines de mode.

Nous souhaitons tester la cohérence entre les prédictions obtenues par analyse qualitative des modèles et les données expérimentales sur la dynamique du système. Etant donné les limitations actuelles des méthodes expérimentales pour la quantification des concentrations cellulaires des molécules d'intérêt, protéines ou ARNm, nous avons choisi d'utiliser seulement l'information qualitative correspondant au sens de variation du signal, et non pas son intensité. Autrement dit, nous supposons que les données expérimentales disponibles donnent des informations fiables sur les sens de variation des concentrations des

protéines du réseau, mais pas nécessairement sur les valeurs de ces concentrations. Cette information est plus pauvre, mais son utilisation repose sur des hypothèses moins fortes. Ainsi, notre méthode ne repose pas sur l'hypothèse que la concentration cellulaire de la molécule d'intérêt est proportionnelle au signal mesuré, mais simplement qu'elle varie dans le même sens.

Toutefois, le niveau d'abstraction utilisé dans l'approche mentionnée ci-dessus ne permet pas de caractériser sans ambiguïté les signes des dérivées des solutions dans les différentes régions de l'espace des phases, et donc, ne permet pas de prédire les sens de variation des concentrations des protéines. Les comportements prédits et observés ne sont donc pas directement comparables. Ce problème peut être résolu en utilisant un niveau d'abstraction moins élevé. Pour cela, nous avons proposé l'utilisation d'une partition plus fine de l'espace des phases, obtenue en repartitionnant les domaines de mode en domaines dits de flot. Le graphe de transitions obtenu par abstraction discrète à partir de la partition plus fine correspond alors à une prédiction du comportement du système qui peut être comparée aux comportements observés. De nouvelles règles pour le calcul du graphe de transitions ont été proposées (chapitre 4 et [25]).

Egalement, étant donné que la taille du graphe de transitions croît fortement avec la taille (nombre de gènes) et la complexité (nombre d'interactions) du réseau, rendant la vérification manuelle des propriétés dynamiques du système difficile et sujette à l'erreur, nous avons proposé d'utiliser des techniques de model checking, qui permettent de vérifier efficacement les propriétés dynamiques des systèmes de transitions discrets. L'utilisation combinée d'abstraction discrète et de model checking permet ainsi de tester efficacement les propriétés dynamiques de modèles continus de réseaux de régulation génique (chapitre 5 et [21, 23]). En particulier, en utilisant le fait que par abstraction discrète on obtient une approximation conservative du système, et en utilisant seulement les fragments universels des logiques temporelles, il est possible de garantir qu'un modèle ne peut être rejeté à tort.

Cette méthode a été implémentée dans une nouvelle version de l'outil *Genetic Network Analyzer*, GNA 6.0 (chapitre 6). Cette implémentation reprend les fonctionnalités de la version précédente d'édition des modèles et d'analyse visuelle des graphes de transitions obtenus, mais réalise une analyse plus fine de la dynamique du système, le graphe de transitions calculé étant basé sur une partition plus fine que précédemment de l'espace des phases. De plus, GNA 6.0 permet l'export des graphes vers deux model checkers, NuSMV et CADP. Il est ainsi possible d'exploiter les avantages offerts (formulation aisée ou expressivité élevée) par différentes approches et différents outils de model checking pour l'analyse des résultats de simulation. Le fait que GNA soit dédié exclusivement à l'analyse d'une classe particulière de modèles par la méthode présentée dans ce travail rend possible diverses optimisations, qui permettent un calcul des prédictions qui supporte bien le passage à l'échelle.

Egalement, cette méthode a été appliquée à la validation de deux modèles d'intérêt biologique, grands et complexes (chapitres 7, et 8 et [24]). Les prédictions obtenues à partir de ces deux modèles ont été comparées avec des données expérimentales disponibles dans la littérature. Nous avons pu vérifier que, dans les deux cas, les prédictions obtenues sont cohérentes avec la plupart des propriétés observées du système. Egalement, plusieurs incohérences ont été identifiées, suggérant que des révisions ou des extensions des modèles sont nécessaires, ou que des expériences complémentaires doivent être réalisées. De plus, par cette méthode des prédictions nouvelles concernant l'existence d'oscillations dans la concentration de certaines protéines, consécutivement à une disponibilité soudaine en

carbone, chez *E. coli*, ont pu être mieux caractérisées.

Enfin, les contributions apportées par ce travail peuvent être synthétisées selon quatre points (voir également chapitre 9).

Premièrement, nous proposons d'utiliser une abstraction discrète en combinaison avec du model checking pour l'analyse de modèles continus de réseaux biologiques. L'utilisation de ces deux techniques offre des avantages complémentaires, qui, ensemble, permettent l'analyse des propriétés dynamiques de modèles continus grands et complexes. Plus précisément, l'abstraction discrète permet de décrire la dynamique du système par un graphe de transitions entre états discrets ayant une signification intuitive et une sémantique mathématiquement bien fondée, et le model checking permet l'analyse efficace des grands graphes obtenus par abstraction discrète.

Deuxièmement, nous utilisons un niveau d'abstraction suffisamment fin pour permettre d'obtenir une description des comportements du système adaptée à la comparaison avec les données expérimentales disponibles (prédiction des sens de variation des concentrations des protéines), mais qui n'implique toutefois pas l'utilisation de procédures de décision complexes pour le calcul des prédictions. La méthode repose sur l'utilisation d'une partition relativement complexe de l'espace des phases, utilisée pour décrire finement l'état du système. Également, des inclusions différentielles sont utilisées pour prendre en compte des phénomènes dynamiques souvent négligés, dus à la présence de discontinuités dans le second membre des équations différentielles et ayant lieu au niveau des seuils de concentrations. L'établissement de règles qui demeurent facile à évaluer pour le calcul symbolique du graphe de transitions a posé des problèmes non triviaux.

Troisièmement, la méthode proposée dans ce travail a été implémentée dans une nouvelle version de l'outil *Genetic Network Analyzer* (GNA 6.0). Son implémentation a nécessité la réécriture d'une partie importante du moteur de calcul du graphe de transitions et l'ajout de fonctionnalités pour l'export des graphes vers deux model checkers. GNA 6.0 est librement disponible pour la recherche académique¹ et fait actuellement l'objet d'un transfert technologique pour sa diffusion gratuite auprès d'utilisateurs académiques et sa commercialisation auprès d'utilisateurs non académiques par la société Genostar Technologies.

Quatrièmement, l'applicabilité de l'approche proposée, et la pertinence biologique des résultats obtenus a été démontrée par deux applications à la validation de modèles d'intérêt biologique grands et complexes. L'approche que nous proposons a également suggéré la réalisation d'un type d'expérience spécialement adaptée à la validation de modèles, dont le but serait de mesurer, avec un taux d'échantillonnage élevé, et en parallèle, l'évolution des concentrations de toutes les protéines du modèle. Les informations obtenues par ce type d'expérience permettraient alors pleinement de mettre à l'épreuve la validité du modèle. Un tel programme expérimental est actuellement en cours pour la validation du modèle de la réponse au stress nutritionnel chez *E. coli*.

10.2 Perspectives

Plusieurs directions peuvent être envisagées pour des travaux futurs. Dans cette section, nous mentionnons deux approfondissements possibles de la méthode présentée permettant d'améliorer son applicabilité, puis nous suggérons différents prolongements envisageables pour traiter des problèmes plus complexes.

¹Contact : hidde.de-jong@inrialpes.fr.

Une direction relativement évidente de travail futur concerne l'amélioration du couplage entre GNA et les model checkers. Avec l'implémentation existante de la méthode, le calcul des prédictions (*i.e.* du graphe de transitions) est réalisé dans l'environnement de GNA, tandis que la vérification des propriétés est réalisée dans l'environnement des model checkers. Etant donné le profil des utilisateurs (biologistes modélisateurs), les différentes étapes de la vérification devraient être intégrées dans GNA, et adaptées en conséquence. Il est, par exemple, envisageable de développer une logique temporelle et un éditeur graphique permettant de formuler dans cette logique les propriétés à vérifier, qui soient adaptés au type de questions habituellement posées pour la validation de ce type de modèles. Également, l'analyse des diagnostics retournés par les model checkers devrait pouvoir être faite dans l'environnement de GNA.

Une autre direction possible de travail futur pour l'amélioration de l'applicabilité de l'approche proposée vise à exploiter des techniques avancées de model checking afin de permettre l'analyse de modèles significativement plus grands et plus complexes que ceux qui sont actuellement analysables par GNA. Parmi les nombreuses techniques développées dans la communauté du model checking, certaines semblent particulièrement prometteuses. Par exemple, l'utilisation d'une représentation symbolique plutôt qu'explicite du graphe de transitions à analyser, notamment à l'aide de diagrammes de décision binaires, permet de diminuer parfois considérablement la taille de l'espace mémoire nécessaire [41]. D'autres techniques permettent de focaliser l'analyse sur les seules informations pertinentes. Parmi celles-ci figurent les techniques utilisant les ordres partiels (« partial order reduction ») [101], qui sont applicables lorsque l'ordre relatif de certains événements n'importe pas, comme par exemple l'ordre d'activation de deux gènes devant tous deux être exprimés dans un même processus, et les techniques de minimisation de graphes basées sur l'équivalence observationnelle [165], qui s'appliquent lorsque seules certaines propriétés sont observables sur le système, comme par exemple lorsque des données ne sont disponibles que pour une protéine (*i.e.* une variable). Enfin, les réseaux de régulation génique ont, ou semblent avoir, une structure et un fonctionnement relativement modulaire [111]. En exploitant cette modularité, il devient envisageable d'analyser de très grands réseaux. Naturellement, dans ce contexte, l'utilisation de techniques de vérification compositionnelle [142] semblent particulièrement bien adaptée.

En dehors de ces améliorations possibles de l'approche proposée dans ce travail, d'autres perspectives peuvent être envisagées, en considérant la vérification de modèles non plus comme un but à atteindre, mais comme un outil disponible. Un prolongement relativement immédiat de la méthode que nous avons développée concerne la sélection de modèles. Ce problème consiste à sélectionner parmi un ensemble de modèles potentiels, les modèles qui satisfont un ensemble de critères donnés (voir également Bernot *et al.* [33] et Calzone *et al.* [43]). Deux problèmes plus complexes peuvent alors être considérés.

Premièrement, il devient envisageable de développer des méthodes pour l'inférence et pour la révision de modèles. Le premier problème consiste à compléter un modèle incomplètement spécifié de telle sorte qu'il satisfasse un ensemble de contraintes. Le second problème consiste à identifier et à éliminer dans des modèles non cohérents avec les contraintes données le, ou les, éléments qui causent cette incohérence. Trouver une méthode efficace pour résoudre ces problèmes semble non trivial. Le développement d'une telle méthode et son implémentation dans un outil d'aide à la modélisation seraient très certainement une contribution majeure pour la compréhension du fonctionnement des réseaux de régulation géniques. Pour apporter des solutions à ces problèmes, il est par exemple envisageable

de s'inspirer soit de techniques d'apprentissage automatique [166], soit de techniques de diagnostique à base de modèles [109].

Un second problème envisageable concerne la conception de modèles. L'objectif est de trouver un modèle qui satisfasse un ensemble donné de contraintes élémentaires et de spécifications du comportement. Les applications naturelles de cette méthode seraient dans le domaine émergent de la biologie synthétique [81], dont l'objectif est de créer des systèmes biologiques ayant un comportement donné. Bien que ce problème soit un cas particulier d'inférence de modèle (à partir d'un modèle très incomplet), l'utilisation de techniques d'analyse et de contrôle de systèmes continus ou hybrides semble plus prometteur pour aborder ce problème.

Sixième partie

Annexes

A Preuves

A.1 Analyse de la dynamique des systèmes LPM

Lemme 1. [44, 70] Pour tout $M \in \mathcal{M}_{pers}$ et $\xi \in \Xi_\Sigma$ dans M , on a

1. si M est un domaine de mode régulier, alors ξ converge monotoniquement vers $\Psi(M)$,
2. si M est un domaine de mode singulier, alors ξ_i converge monotoniquement vers $\Psi_i(M)$ pour tout $i \in \{1, \dots, n\}$ tel que $\xi_i(0) \notin \Psi_i(M)$.

Démonstration. Soient M un domaine de mode persistant et $\xi \in \Xi_\Sigma$ une solution dans M . Si M est un domaine de mode régulier, alors $\Psi(M) = \{\psi(M)\}$ (définition 13) et ξ est donné par : $\xi(t) = \psi(M) - (\psi(M) - \mathbf{x}_0) e^{-\nu^M t}$, pour un certain $\mathbf{x}_0 \in M$. Il est clair que ξ converge monotoniquement vers $\psi(M)$ tant qu'elle reste dans M .

Si M n'est pas un domaine de mode régulier, alors c'est un domaine de mode singulier, et $\Psi(M)$ peut être un ensemble non réduit à un point. Considérons $i \in \{1, \dots, n\}$ tel que $\xi_i(0) \notin \Psi_i(M)$. Puisque M est persistant, $\Psi(M) \neq \emptyset$ et d'après la définition 14, $\Psi_i(M) = [\min_{M' \in R(M)} \psi_i(M'), \max_{M' \in R(M)} \psi_i(M')]$. Supposons que l'on ait $\xi_i(t) < \min \Psi_i(M)$ pour $t \in [0, \tau]$, avec $\tau \in \mathbb{R}_{>0} \cup \{\infty\}$. On a alors $\xi_i(t) < \psi_i(M') = \mu_i^{M'} / \nu_i^{M'}$, pour tout $M' \in R(M)$. D'après (4.2), $\min H_i(\xi(t)) = \min_{M' \in R(M)} \{\mu_i^{M'} - \nu_i^{M'} \xi_i(t)\}$. Donc $\min H_i(\xi(t)) > 0$ et, puisque $\dot{\xi}_i(t) \in H_i(\xi(t))$ est vrai presque partout sur $[0, \tau]$, on a presque partout sur $[0, \tau]$, $\dot{\xi}_i(t) > 0$, et ξ_i est strictement croissante sur $[0, \tau]$. Par conséquent, ξ_i converge monotoniquement vers $\Psi_i(M)$ tant qu'elle reste dans M . Le cas $\xi_i(t) > \max \Psi_i(M)$ est démontré de façon similaire. ■

Lemme 2. Pour tout $M \in \mathcal{M}_{pers}$, $\mathbf{x} \in M$, et $\psi \in \Psi(M)$, il existe une solution $\xi \in \Xi_\Sigma$ dans M telle que $\xi(t_x) = \mathbf{x}$ pour un certain $t_x \geq 0$, et ξ_i converge monotoniquement vers ψ_i , pour tout $i \in \{1, \dots, n\}$.

Démonstration. Soient M un domaine de mode persistant, \mathbf{x} un point dans M , et ψ un point dans $\Psi(M)$. Si M est un domaine de mode régulier, alors d'après la définition 13, on a $\Psi(M) = \{\psi\}$. Par conséquent, il existe une unique solution $\xi \in \Xi_\Sigma$ telle que $\xi(t_x) = \mathbf{x}$ pour un instant $t_x \geq 0$ donné : $\xi(t) = \psi - (\psi - \mathbf{x})e^{-\nu^M(t-t_x)}$. Il est clair que pour tout $i \in \{1, \dots, n\}$, ξ_i converge monotoniquement vers ψ_i tant que ξ reste dans M (voir également le lemme 1).

Si M n'est pas un domaine de mode régulier, alors c'est un domaine de mode singulier. Soit ξ la fonction absolument continue définie comme suit. Si $x_i = \psi_i$, $i \in \{1, \dots, n\}$, alors $\xi_i(t) = x_i$, pour tout $t \geq 0$. Sinon, supposons que $x_i < \psi_i$ (le cas $x_i > \psi_i$ est similaire). D'après la définition 14 et le fait que M est persistant, comme pour le lemme 1, nous avons $x_i < \psi_i(M') = \mu_i^{M'} / \nu_i^{M'}$, pour un certain $M' \in R(M)$. Posons alors $\xi_i(t) = \psi_i(M') + (x_i - \psi_i(M'))e^{-\nu_i^{M'}(t-t_x)}$, si $\xi_i(t) < \psi_i$, et $\xi_i(t) = \psi_i$, sinon. $\xi = (\xi_1, \dots, \xi_n)'$ satisfait $\xi(t_x) = \mathbf{x}$, et $\dot{\xi}(t) \in H(\xi(t))$ est vrai presque partout tant que ξ reste dans M . Par conséquent, ξ est une solution de (3.8) dans M (définition 11). De plus tant que ξ reste dans M , et pour tout $i \in \{1, \dots, n\}$, ξ_i converge monotoniquement vers ψ_i . ■

A.2 Abstraction discrète et modèles qualitatifs

Proposition 2 (Atteignabilité équivalente). Pour tout $\mathbf{x}, \mathbf{x}' \in \Omega$, il existe une solution ξ de Σ et τ, τ' tels que $0 \leq \tau \leq \tau'$, $\xi(\tau) = \mathbf{x}$, et $\xi(\tau') = \mathbf{x}'$ si et seulement si il existe une exécution $(\mathbf{x}^0, \dots, \mathbf{x}^m)$ de Σ -TS telle que $\mathbf{x}^0 = \mathbf{x}$ et $\mathbf{x}^m = \mathbf{x}'$.

Démonstration.

(Sens direct) Considérons deux points $\mathbf{x}, \mathbf{x}' \in \Omega$, une solution $\xi \in \Xi_\Sigma$ et τ, τ' tels que $0 \leq \tau \leq \tau'$, $\xi(\tau) = \mathbf{x}$, et $\xi(\tau') = \mathbf{x}'$. Si $\tau = \tau'$, alors $(\xi(\tau))$ est une exécution triviale satisfaisant les conditions de la proposition 2. Sinon, $\tau < \tau'$ et nous notons D^0, \dots, D^m , la séquence, ordonnée dans le temps, de domaines de flots traversés par ξ pendant l'intervalle de temps $[\tau, \tau']$. D^0, \dots, D^m est une séquence finie, puisque, par définition, toutes les solutions dans Ξ_Σ atteignent ou quittent un hyperplan de seuil un nombre fini de fois (voir section 3.2). Si $m = 0$, alors d'après la définition 18, il existe une transition de type *int* de $\xi(\tau)$ vers $\xi(\tau')$, et par conséquent, $(\xi(\tau), \xi(\tau'))$ est une exécution satisfaisant les conditions de la proposition 2. Sinon, $m > 0$, et nous notons $\sigma^0, \dots, \sigma^{m-1}$, les instants où ξ passe d'un domaine à un autre, définis formellement comme $\sigma^i = \sup \{t \in [\tau, \tau'] \mid \xi(t) \in D^i\}$. Enfin, nous définissons une séquence d'instants τ^0, \dots, τ^m telle que $\xi(\tau^i) \in D^i$, pour tout $i \in \{0, \dots, m\}$. Plus précisément, nous définissons τ^0 comme τ , τ^i comme $(\sigma^{i-1} + \sigma^i)/2$, pour tout $i \in \{1, \dots, m-1\}$, et τ^m comme τ' . Alors nous prouvons par récurrence sur i que $(\xi(\tau^0), \dots, \xi(\tau^i))$ est une exécution, pour tout $i \in \{0, \dots, m\}$. L'hypothèse de récurrence est clairement vraie pour $i = 0$. Supposons que $(\xi(\tau^0), \dots, \xi(\tau^i))$ est une exécution, pour un certain $i \in \{0, \dots, m-1\}$. Remarquons tout d'abord qu'une fonction absolument continue ne peut pas traverser instantanément deux domaines de flot consécutifs. En conséquence, nous distinguons seulement trois cas, selon que ξ traverse instantanément D^i, D^{i+1} , ou aucun des deux. Dans le premier cas, on a $\tau^i = \sigma^i < \tau^{i+1}$. Donc ξ quitte D^i à l'instant τ^i et reste dans D^{i+1} durant l'intervalle de temps $(\tau^i, \tau^{i+1}]$. Alors, d'après la définition 18, il existe une transition de type *dim*⁺ de $\xi(\tau^i)$ vers $\xi(\tau^{i+1})$. Dans le second cas, on a $\tau^i < \sigma^{i+1} = \tau^{i+1}$. On montre alors facilement que cette situation correspond à une transition de type *dim*⁻ de $\xi(\tau^i)$ vers $\xi(\tau^{i+1})$. Dans le dernier cas, on a $\tau^i < \sigma^i < \tau^{i+1}$. Cette situation correspond à une transition de type *int* (*dim*⁻) de $\xi(\tau^i)$ vers $\xi(\sigma^i)$, suivie d'une transition de type *dim*⁺ (*int*) de $\xi(\sigma^i)$ vers $\xi(\tau^{i+1})$, si $\xi(\sigma^i) \in D^i$ ($\xi(\sigma^i) \in D^{i+1}$). Dans tous les cas, nous pouvons conclure que $(\xi(\tau^0), \dots, \xi(\tau^{i+1}))$ est une exécution de Σ -TS.

(Réciproque) Considérons une exécution $(\mathbf{x}^0, \dots, \mathbf{x}^m)$ de Σ -TS, avec $\mathbf{x}^0 = \mathbf{x}$ et $\mathbf{x}^m = \mathbf{x}'$. Si $m = 0$, alors $\mathbf{x} = \mathbf{x}'$ et toute solution $\xi \in \Xi_\Sigma$, satisfaisant $\xi(0) = \mathbf{x}$ satisfait les conditions de la proposition 2, avec $\tau = \tau' = 0$. Dans la suite, nous supposons que $m > 0$. Alors, d'après la définition 18, il existe une séquence de solutions, $(\xi^i)_{i \in \{0, \dots, m-1\}}$, et une séquence d'instants, $(\tau^i)_{i \in \{0, \dots, m\}}$, telles que ξ^i est définie sur l'intervalle de temps $[\tau^i, \tau^{i+1}]$, avec $\tau^i < \tau^{i+1}$, et satisfait $\xi^i(\tau^i) = \mathbf{x}^i$ et $\xi^i(\tau^{i+1}) = \mathbf{x}^{i+1}$, $i \in \{0, \dots, m-1\}$. Alors nous prouvons que la concaténation de ces solutions forme une solution qui satisfait les conditions de la proposition 2. Définissons la fonction constante par morceau $I : [\tau^0, \tau^m] \rightarrow \{0, \dots, m-1\}$, telle que $I(t) = \{i \mid t \in [\tau^i, \tau^{i+1}]\}$, pour tout $t \in [\tau^0, \tau^m]$, et $I(\tau^m) = m-1$. Alors nous définissons la concaténation ξ des solutions ξ^i comme $\xi(t) = \xi^{I(t)}(t)$ sur l'intervalle de temps $[\tau^0, \tau^m]$. Notons tout d'abord que $\xi(\tau^0) = \xi^0(\tau^0) = \mathbf{x}$ et que $\xi(\tau^m) = \xi^{m-1}(\tau^m) = \mathbf{x}'$. Également, la continuité absolue de chaque solution ξ^i sur $[\tau^i, \tau^{i+1}]$, avec la continuité de ξ en τ^i , $i \in \{1, \dots, m-1\}$, implique la continuité absolue de ξ sur $[\tau^0, \tau^m]$. De plus, ξ^i satisfait (3.9) presque partout sur $[\tau^i, \tau^{i+1}]$, $i \in \{0, \dots, m-1\}$, de telle sorte que ξ satisfait (3.9) presque partout sur $[\tau^0, \tau^m]$. Enfin, puisque, d'après la définition 18,

la solution ξ^i passe d'un domaine à un autre au plus une fois durant l'intervalle de temps $[\tau^i, \tau^{i+1}]$, ξ traverse au plus m domaines de flot durant l'intervalle de temps $[\tau^0, \tau^m]$. Donc ξ est une solution de Σ qui satisfait les conditions de la proposition 2. ■

A.3 Calcul symbolique du système de transitions LPM qualitatif

Proposition 5 (Calcul de *Dsign* : domaine instantané). Soit $D \in \mathcal{D}$ un domaine de flot instantané et $S \in 2^{\{-1,0,1\}^n}$ tel que $D \models_{\sim\Omega} Dsign = S$. Alors, $S = \emptyset$.

Démonstration. Soient D un domaine de flot instantané et \mathbf{x} un point dans D . Puisque D est un domaine de flot instantané, aucune solution ne reste dans D . Par conséquent, l'ensemble des solutions $\xi \in \Xi_\Sigma$ dans D et passant en \mathbf{x} est vide, et d'après la définition 17, $S(\mathbf{x}) = \emptyset$. Ceci implique que $S = \emptyset$ (proposition 3, section 4.2.2). ■

Proposition 6 (Calcul des transitions *int*). Soient D et D' deux domaines de flot. $D \xrightarrow{int}_{\sim\Omega} D'$ si et seulement si

1. $D = D'$, et
2. $D \in \mathcal{D}_{pers}$

Démonstration.

(Sens direct) D'après la proposition 3 (section 4.2.2), si il existe une transition de type *int* de D vers D' , alors $D = D'$, et au moins une solution ne traverse pas D instantanément. Donc, D est un domaine de flot persistant.

(Réciproque) Par définition d'un domaine de flot persistant D (section 4.1.3), il existe au moins une solution ne traversant pas D instantanément. Alors, cette solution satisfait les conditions de la proposition 3 (section 4.2.2) pour l'existence d'une transition de type *int* de D vers D' , avec $D = D'$. ■

Proposition 8 (Calcul des transitions *dim⁻*). Soient D et D' deux domaines de flot. $D \xrightarrow{dim^-}_{\sim\Omega} D'$ si et seulement si

1. $D' \subseteq \partial D$, et
2. il existe $\mathbf{x} \in D$, $\mathbf{x}' \in D'$, et $\psi \in \Psi(mode(D))$, tels que
 - (a) pour tout $i \in \{1, \dots, n\}$ pour lesquels D'_i , mais pas D_i , est un singleton, on a

$$(\psi_i - x'_i)(x'_i - x_i) > 0,$$

ou (b) on a

$$\mathbf{x}' = \psi.$$

Démonstration. ¹

(Sens direct) D'après la proposition 3, une transition de type *dim⁻* de D vers D' implique l'existence d'une solution ξ absolument continue sur l'intervalle de temps $[\tau, \tau']$ satisfaisant $\xi(t) \in D$ pour tout $t \in [\tau, \tau')$, et $\xi(\tau') \in D'$. Alors, par le même raisonnement que dans la preuve précédente, on montre que $D' \subseteq \partial D$. Donc la condition 1 doit être satisfaite.

¹Des preuves similaires sont données dans [44, 70].

Supposons que la condition 2 ne soit pas satisfaite. Autrement dit, nous supposons que (a) il existe $i \in \{1, \dots, n\}$ tel que D'_i , mais pas D_i , est un singleton, et tel que $(\psi_i - x'_i)(x'_i - x_i) \leq 0$ pour tout $\mathbf{x} \in D$, $\mathbf{x}' \in D'$, et $\boldsymbol{\psi} \in \Psi(\text{mode}(D))$, et que (b) $D' \cap \Psi(\text{mode}(D)) = \emptyset$. Supposons que $D'_i = \{x'_i \in \Omega_i \mid x'_i = \lambda_i\}$, pour un certain $\lambda_i \in \Lambda_i \cup \Theta_i$. Si $x'_i < x_i$, alors (a) implique que $\psi_i \geq x'_i$. De façon similaire, si $x'_i > x_i$, alors $\psi_i \leq x'_i$. Si pour tout $\boldsymbol{\psi} \in \Psi(\text{mode}(D))$ les inégalités précédentes sont strictes, alors, comme dans la preuve de la proposition 7, nous pouvons utiliser le lemme 1 et en déduire qu'aucune solution ne peut atteindre D' à partir de D . Sinon, on montre que les solutions $\boldsymbol{\xi}$ ne peuvent atteindre l'hyperplan $\{\mathbf{x}' \in \Omega \mid x'_i = \lambda_i\}$ qu'asymptotiquement. Puisque $\boldsymbol{\xi}$ dans D converge vers $\Psi(\text{mode}(D))$, si une solution $\boldsymbol{\xi}$ atteint $D' \subset \{\mathbf{x}' \in \Omega \mid x'_i = \lambda_i\}$, cela implique que $D' \cap \Psi(\text{mode}(D))$ n'est pas vide, ce qui est en contradiction avec (b). La condition 2 doit donc être satisfaite.

(Réciproque) Supposons que soit les conditions 1 et 2(a), soit les conditions 1 et 2(b) sont vérifiées. Dans les deux cas, $\Psi(\text{mode}(D)) \neq \emptyset$, de telle sorte que $\text{mode}(D)$, et donc D , sont persistants. Premièrement, supposons que les conditions 1 et 2(a) sont vérifiées. Soit $i \in \{1, \dots, n\}$ tel que D'_i , mais pas D_i , est un singleton. Alors, $D'_i = \{x'_i \in \Omega_i \mid x'_i = \lambda_i\}$, pour un certain $\lambda_i \in \Lambda_i \cup \Theta_i$. Il découle de la condition 2(a) qu'il existe $\mathbf{x} \in D$ et $\boldsymbol{\psi} \in \Psi(\text{mode}(D))$, tels que $(\psi_i - x'_i)(x'_i - x_i) > 0$. Plus précisément, nous supposons que $\boldsymbol{\psi} \in \Psi(\text{mode}(D)) \cap \text{supp}(D)$, l'existence d'un tel $\boldsymbol{\psi}$ découlant de la persistance de D et du lemme 1. Supposons que $x'_i < x_i$. Etant donné la définition des domaines de flot, cette inégalité est vraie pour tout $\mathbf{x} \in D$. De plus, on a $\psi_i < x'_i$. Donc, pour tout $\mathbf{x} \in D$ et pour un certain $\boldsymbol{\psi} \in \Psi(\text{mode}(D)) \cap \text{supp}(D)$, on a $\psi_i < x'_i < x_i$. Symétriquement, si il existe un certain $\mathbf{x} \in D$ tel que $x'_i > x_i$, alors pour tout $\mathbf{x} \in D$ et pour un certain $\boldsymbol{\psi} \in \Psi(\text{mode}(D)) \cap \text{supp}(D)$, on a $\psi_i > x'_i > x_i$. D'après le lemme 2, et puisque $\boldsymbol{\psi} \in \Psi(\text{mode}(D)) \cap \text{supp}(D)$, il existe, pour tout $\mathbf{x} \in D$, une solution $\boldsymbol{\xi}$ dans D passant par \mathbf{x} et convergeant monotoniquement vers $\boldsymbol{\psi}$. Par conséquent, ces solutions, atteignent en temps fini un ou plusieurs hyperplans $\{\mathbf{x}' \in \Omega \mid x'_i = \lambda_i\}$, $i \in I$, avec I l'ensemble des indices i tels que D'_i , mais pas D_i , est un singleton. On peut choisir $\mathbf{x} \in D$ tel que la solution correspondante $\boldsymbol{\xi}$ passe par $D' \subset \cap_{i \in I} \{\mathbf{x}' \in \Omega \mid x'_i = \lambda_i\}$. On vérifie alors facilement qu'une telle solution $\boldsymbol{\xi}$ satisfait les conditions de la proposition 3 pour l'existence d'une transition de type dim^- .

Deuxièmement, supposons que les conditions 1 et 2(b) sont vérifiées. Considérons $\boldsymbol{\psi} \in \Psi(\text{mode}(D)) \cap D'$ et une solution $\boldsymbol{\xi}$ dans $\text{mode}(D)$, donnée par le lemme 2, qui part de D et converge monotoniquement vers $\boldsymbol{\psi}$. Puisque $D' \subseteq \partial D$, on a $\boldsymbol{\psi} \in D' \subseteq \text{supp}(D)$, et par conséquent, $\boldsymbol{\xi}$ est une solution dans D . $\boldsymbol{\xi}$ atteint asymptotiquement $\boldsymbol{\psi}$, et par conséquent D' . Donc, $\boldsymbol{\xi}$ satisfait les conditions de la proposition 3 pour l'existence d'une transition de type dim^- . ■

B Exemples de fichiers

B.1 Modèles et conditions initiales pour GNA

Représentation textuelle, dans le format `.gna`, du modèle du réseau d'inhibition réciproque présenté dans la partie II. Les équations d'état correspondent à (3.4)-(3.5), et les contraintes d'inégalité sur les paramètres, à (4.3)-(4.4).

```

/*
xaxb.gna
two-gene model used as running example in HSCC05, IJCAI05, and ISMB05
*/
state-variable: xa
  zero-parameter: z_xa
  box-parameter: max_xa
  threshold-parameters: t_xa1, t_xa2
  synthesis-parameters: k_xa
  degradation-parameters: g_xa
  state-equation:
    d/dt xa = k_xa * s-(xa, t_xa2) * s-(xb,t_xb) - g_xa * xa
  parameter-inequalities:
    z_xa < t_xa1 < t_xa2 < k_xa/g_xa < max_xa ;
state-variable: xb
  zero-parameter: z_xb
  box-parameter: max_xb
  threshold-parameters: t_xb
  synthesis-parameters: k_xb
  degradation-parameters: g_xb
  state-equation:
    d/dt xb = k_xb * s-(xa, t_xa1) - g_xb * xb
  parameter-inequalities:
    z_xb < t_xb < k_xb/g_xb < max_xb ;

```

Représentation textuelle, dans le format `.dat`, de conditions initiales pour le modèle du réseau d'inhibition réciproque présenté dans la partie II. Ces conditions correspondent à un état dans lequel les concentrations des deux protéines sont basses, et plus spécifiquement, à l'état $D^{1.1}$ représenté figure 4.8.

```

/*
xaLow_xbLow.dat
to be used with two-gene model xaxb.gna
*/
initial-conditions:
xa >= z_xa; xa < t_xa1;
xb >= z_xb; xb < t_xb;

```

B.2 Exports vers NuSMV

Fichier d'export du graphe de transitions obtenu figure 6.3, pour le réseau d'inhibition réciproque, et donné dans le langage d'entrée de NuSMV (format `.smv`), pour une analyse par le model checker NuSMV.

```

MODULE main
VAR
  QState_number : 1..9;
  min_xa : {GE_z_xa, GT_z_xa, GE_t_xa1, GT_t_xa1, GE_t_xa2, GT_t_xa2, GE_k_xa_DIV_g_xa,
           GT_k_xa_DIV_g_xa};
  max_xa : {LE_z_xa, LT_t_xa1, LE_t_xa1, LT_t_xa2, LE_t_xa2, LT_k_xa_DIV_g_xa, LE_k_xa_DIV_g_xa,
           LE_max_xa};
  DSignOfxa : {_DEC_, _STD_, _INC_, _ALL_, _UNDEF_};
  min_xb : {GE_z_xb, GT_z_xb, GE_t_xb, GT_t_xb, GE_k_xb_DIV_g_xb, GT_k_xb_DIV_g_xb};
  max_xb : {LE_z_xb, LT_t_xb, LE_t_xb, LT_k_xb_DIV_g_xb, LE_k_xb_DIV_g_xb, LE_max_xb};
  DSignOfxb : {_DEC_, _STD_, _INC_, _ALL_, _UNDEF_};

DEFINE
  inQS1 := QState_number = 1
    & min_xa = GE_z_xa
    & max_xa = LT_t_xa1
    & min_xb = GE_z_xb
    & max_xb = LT_t_xb
    & DSignOfxa= _INC_
    & DSignOfxb= _INC_;
  inQS2 := QState_number = 2
    & min_xa = GE_t_xa1
    & max_xa = LE_t_xa1
    & min_xb = GT_z_xb
    & max_xb = LT_t_xb
    & DSignOfxa= _UNDEF_
    & DSignOfxb= _UNDEF_;
  inQS3 := QState_number = 3
    & min_xa = GT_z_xa
    & max_xa = LT_t_xa1
    & min_xb = GE_t_xb
    & max_xb = LE_t_xb
    & DSignOfxa= _UNDEF_
    & DSignOfxb= _UNDEF_;
  inQS4 := QState_number = 4
    & min_xa = GE_t_xa1
    & max_xa = LE_t_xa1
    & min_xb = GE_t_xb
    & max_xb = LE_t_xb
    & DSignOfxa= _STD_
    & DSignOfxb= _STD_;
  inQS5 := QState_number = 5
    & min_xa = GT_t_xa1
    & max_xa = LT_t_xa2
    & min_xb = GT_z_xb
    & max_xb = LT_t_xb
    & DSignOfxa= _INC_
    & DSignOfxb= _DEC_;
  inQS6 := QState_number = 6
    & min_xa = GT_z_xa
    & max_xa = LT_t_xa1
    & min_xb = GT_t_xb

```

```

    & max_xb = LT_k_xb_DIV_g_xb
    & DSignOfxa= _DEC_
    & DSignOfxb= _INC_;
inQS7 := QState_number = 7
    & min_xa = GE_z_xa
    & max_xa = LE_z_xa
    & min_xb = GE_k_xb_DIV_g_xb
    & max_xb = LE_k_xb_DIV_g_xb
    & DSignOfxa= _STD_
    & DSignOfxb= _STD_;
inQS8 := QState_number = 8
    & min_xa = GE_t_xa2
    & max_xa = LE_t_xa2
    & min_xb = GT_z_xb
    & max_xb = LT_t_xb
    & DSignOfxa= _STD_
    & DSignOfxb= _DEC_;
inQS9 := QState_number = 9
    & min_xa = GE_t_xa2
    & max_xa = LE_t_xa2
    & min_xb = GE_z_xb
    & max_xb = LE_z_xb
    & DSignOfxa= _STD_
    & DSignOfxb= _STD_;
isInitialState := inQS1;
isSteadyState := inQS7 | inQS9 | inQS4;

INIT
    inQS1;

TRANS
    case
        QState_number = 1 : next(inQS1 | inQS2 | inQS3 | inQS4);
        QState_number = 2 : next(inQS5);
        QState_number = 3 : next(inQS6);
        QState_number = 4 : next(inQS4 | inQS5 | inQS6);
        QState_number = 5 : next(inQS5 | inQS8);
        QState_number = 6 : next(inQS6 | inQS7);
        QState_number = 7 : next(inQS7);
        QState_number = 8 : next(inQS8 | inQS9);
        QState_number = 9 : next(inQS9);
    esac

```


B.3 Exports vers CADP

Fichier d'export du graphe de transitions obtenu figure 6.3, pour le réseau d'inhibition réciproque, et donné dans le format de CADP (format `.aut`), pour une analyse par les outils de CADP.

```

des (0, 18, 10)
(0, "init", 1)
(1, "S1: xa:[z_xa,t_xa1[,inc; xb:[z_xb,t_xb[,inc; initial ", 1)
(1, "S1: xa:[z_xa,t_xa1[,inc; xb:[z_xb,t_xb[,inc; initial ", 2)
(1, "S1: xa:[z_xa,t_xa1[,inc; xb:[z_xb,t_xb[,inc; initial ", 3)
(1, "S1: xa:[z_xa,t_xa1[,inc; xb:[z_xb,t_xb[,inc; initial ", 4)
(2, "S2: xa:{t_xa1},ins; xb:]z_xb,t_xb[,ins; ", 5)
(3, "S3: xa:]z_xa,t_xa1[,ins; xb:{t_xb},ins; ", 6)
(4, "S4: xa:{t_xa1},std; xb:{t_xb},std; equilibrium ", 4)
(4, "S4: xa:{t_xa1},std; xb:{t_xb},std; equilibrium ", 5)
(4, "S4: xa:{t_xa1},std; xb:{t_xb},std; equilibrium ", 6)
(5, "S5: xa:]t_xa1,t_xa2[,inc; xb:]z_xb,t_xb[,dec; ", 5)
(5, "S5: xa:]t_xa1,t_xa2[,inc; xb:]z_xb,t_xb[,dec; ", 8)
(6, "S6: xa:]z_xa,t_xa1[,dec; xb:]t_xb,k_xb/g_xb[,inc; ", 6)
(6, "S6: xa:]z_xa,t_xa1[,dec; xb:]t_xb,k_xb/g_xb[,inc; ", 7)
(7, "S7: xa:{z_xa},std; xb:{k_xb/g_xb},std; equilibrium ", 7)
(8, "S8: xa:{t_xa2},std; xb:]z_xb,t_xb[,dec; ", 8)
(8, "S8: xa:{t_xa2},std; xb:]z_xb,t_xb[,dec; ", 9)
(9, "S9: xa:{t_xa2},std; xb:{z_xb},std; equilibrium ", 9)

```

Bibliographie

References

- [1] M. Adélaïde and G. Sutre. Parametric analysis and abstraction of genetic regulatory networks. In *Proceedings of the Second Workshop on Concurrent Models in Molecular Biology, BioConcur'04*, 2004.
- [2] R. Aebersold and D.R. Goodlett. Mass spectrometry in proteomics. *Chemical Reviews*, 101(2):269–296, 2001.
- [3] E. Alm and A.P. Arkin. Biological networks. *Current Opinion in Structural Biology*, 13(2):193–202, 2003.
- [4] R. Alur, C. Coucoubetis, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.
- [5] R. Alur, C. Courcoubetis, T.A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *First International Workshop on Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 209–229. Springer, 1993.
- [6] R. Alur, T. Dang, and F. Ivancic. Reachability analysis of hybrid systems via predicate abstraction. *ACM Transactions on Embedded Computing Systems*, 2005. To appear.
- [7] R. Alur and D.L. Dill. Automata for modeling real-time systems. In M. Paterson, editor, *Proceedings of the Seventeenth International Colloquium on Automata, Languages and Programming, ICALP'90*, volume 443 of *Lecture Notes in Computer Science*, pages 322–335. Springer, 1990.
- [8] R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(1):183–235, 1994.
- [9] R. Alur, T.A. Henzinger, G. Lafferriere, and G.J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7):971–984, 2000.
- [10] H.R. Andersen. Model checking and boolean graphs. *Theoretical Computer Science*, 126(1):3–30, 1994.
- [11] M. Antoniotti, C. Piazza, A. Policriti, M. Simeoni, and B. Mishra. Taming the complexity of biochemical models through bisimulation and collapsing: Theory and practice. *Theoretical Computer Science*, 325(1):45–67, 2004.
- [12] J.A. Appleman, W. Ross, J. Salomon, and R.L. Gourse. Activation of *Escherichia coli* rRNA transcription by Fis during a growth cycle. *Journal of Bacteriology*, 180(6):1525–32, 1998.

- [13] K. R. Apt, N. Francez, and S. Katz. Appraising fairness in distributed languages. In *Proceedings of the Fourteenth Symposium on Principles of Programming Languages, POPL'87*, pages 189–198. ACM Press, 1987.
- [14] E. Asarin and T. Dang. Abstraction by projection and application to multi-affine systems. In R. Alur and G.J. Pappas, editors, *Seventh International Workshop on Hybrid Systems: Computation and Control, HSCC'04*, volume 2993 of *Lecture Notes in Computer Science*, pages 32–47. Springer, 2004.
- [15] T. Ali Azam, A. Iwata., A. Nishimura, S. Ueda, and A. Ishihama. Growth phase-dependent variation in protein composition of the *Escherichia coli* nucleoid. *Journal of Bacteriology*, 181(20):6361–6370, 1999.
- [16] U. Bai and I. Mandić-Mulec. SinI modulates the activity of SinR, a developmental switch protein of *Bacillus subtilis*, by protein-protein interaction. *Genes and Development*, 7(1):139–148, 1993.
- [17] V.L. Balke and J.D. Gralla. Changes in the linking number of supercoiled DNA accompany growth transitions in *Escherichia coli*. *Journal of Bacteriology*, 169(10):4499–4506, 1987.
- [18] C.A. Ball, R. Osuna, K.C. Ferguson, and R.C. Johnson. Dramatic changes in Fis levels upon nutrient upshift in *Escherichia coli*. *Journal of Bacteriology*, 174(24):8043–8056, 1992.
- [19] A. Balluchi, L. Benvenuti, M.D. Di Benedetto, C. Pinello, and A.L. Sangiovanni-Vincentelli. Automotive engine control and hybrid systems: Challenges and opportunities. *Proceedings of the IEEE*, 88(7):888–911, 2000.
- [20] A. Balluchi, F. Di Natale, A.L. Sangiovanni-Vincentelli, and J.H. van Schuppen. Synthesis for idle speed control of an automotive engine. In R. Alur and G.J. Pappas, editors, *Seventh International Workshop on Hybrid Systems: Computation and Control, HSCC'04*, volume 2993 of *Lecture Notes in Computer Science*, pages 80–94. Springer, 2004.
- [21] G. Batt, D. Bergamini, H. de Jong, H. Garavel, and R. Mateescu. Model checking genetic regulatory networks using GNA and CADP. In S. Graf and L. Mounier, editors, *Eleventh International SPIN Workshop on Model Checking Software*, volume 2989 of *Lecture Notes in Computer Science*, pages 158–163. Springer, 2004.
- [22] G. Batt, R. Casey, H. de Jong, J. Geiselman, J.-L. Gouzé, M. Page, D. Ropers, T. Sari, and D. Schneider. Qualitative analysis of the dynamics of genetic regulatory networks using piecewise-linear models. In E. Pecou, S. Martinez, and A. Maass, editors, *Mathematical and Computational Methods in Biology*. Hermann, 2005. In press.
- [23] G. Batt, D. Ropers, H. de Jong, J. Geiselman, R. Mateescu, M. Page, and D. Schneider. Analysis and verification of qualitative models of genetic regulatory networks : A model-checking approach. In L.P. Kaelbling and A. Saffiotti, editors, *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, IJCAI'05*, pages 370–375, 2005.

- [24] G. Batt, D. Ropers, H. de Jong, J. Geiselman, R. Mateescu, M. Page, and D. Schneider. Validation of qualitative models of genetic regulatory networks by model checking : Analysis of the nutritional stress response in *Escherichia coli*. *Bioinformatics*, 21(Suppl.1):i19–i28, 2005.
- [25] G. Batt, D. Ropers, H. de Jong, J. Geiselman, M. Page, and D. Schneider. Qualitative analysis and verification of hybrid models of genetic regulatory networks: Nutritional stress response in *Escherichia coli*. In M. Morari and L. Thiele, editors, *Eighth International Workshop on Hybrid Systems: Computation and Control, HSCC'05*, volume 3414 of *Lecture Notes in Computer Science*, pages 134–150. Springer, 2005.
- [26] A.M. Bayen, R.L. Raffard, and C.J. Tomlin. Network congestion alleviation using adjoint hybrid control: Application to highways. In R. Alur and G.J. Pappas, editors, *Seventh International Workshop on Hybrid Systems: Computation and Control, HSCC'04*, volume 2993 of *Lecture Notes in Computer Science*, pages 95–110. Springer, 2004.
- [27] G. Behrmann, A. David, K.G. Larsen, O. Möller, P. Pettersson, and W. Yi. UPPAAL - Present and future. In *Proceedings of the Fortieth Conference on Decision and Control, CDC'01*, 2001.
- [28] C. Belta, J. Esposito, J. Kim, and V. Kumar. Computational techniques for analysis of genetic network dynamics. *International Journal of Robotics Research*, 24:219–235, 2005.
- [29] C. Belta, P. Finin, L.C.G.J.M. Habets, A.M. Halász, M. Imieliński, V. Kumar, and H. Rubin. Understanding the bacterial stringent response using reachability analysis of hybrid systems. In R. Alur and G.J. Pappas, editors, *Seventh International Workshop on Hybrid Systems: Computation and Control, HSCC'04*, volume 2993 of *Lecture Notes in Computer Science*, pages 111–125. Springer, 2004.
- [30] B. Bérard, M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci, P. Schnoebelen, and P. McKenzie. *Systems and Software Verification: Model-Checking Techniques and Tools*. Springer, 2001.
- [31] O. Bernard and J.-L. Gouzé. Global qualitative description of a class of nonlinear dynamical systems. *Artificial Intelligence*, 136:29–59, 2002.
- [32] O. Bernard and S. Souissi. Qualitative behavior of stage-structured populations: Application to structural validation. *Journal of Mathematical Biology*, 37:291–308, 1998.
- [33] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. Application of formal methods to biological regulatory networks: Extending Thomas' asynchronous logical approach with temporal logic. *Journal of Theoretical Biology*, 229(3):339–347, 2004.
- [34] N.S. Bjørner, Z. Manna, H. Sipma, and T.E. Uribe. Deductive verification of real-time systems using STeP. In M. Bertran and T. Rus, editors, *Proceedings of the Fourth International AMAST Workshop on Real-Time Systems and Concurrent and Distributed Software, ARTS'97*, volume 1231 of *Lecture Notes in Computer Science*, pages 22–43. Springer, 1997.

- [35] A. Bouajjani, J.-C. Fernandez, and N. Halbwachs. Minimal model generation. In E.M. Clarke and R.P. Kurshan, editors, *Proceedings of the Second International Conference on Computer Aided Verification, CAV'90*, volume 531 of *Lecture Notes in Computer Science*, pages 197–203. Springer, 1990.
- [36] J.C. Bradfield and C. Stirling. Modal logics and mu-calculi: An introduction. In J.A. Bergstra, A. Ponse, and S.A. Smolka, editors, *Handbook of Process Algebra*, pages 293–332. Elsevier, 2001.
- [37] M.S. Branicky, V.S. Borkar, and S.K. Mitter. A unified framework for hybrid control: Model and optimal control theory. *IEEE Transactions on Automatic Control*, 43:31–45, 1998.
- [38] M.S. Branicky, E. Dolginova, and N.A. Lynch. A toolbox for proving and maintaining hybrid specifications. In P.J. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors, *Proceedings of Hybrid Systems IV*, volume 1273 of *Lecture Notes in Computer Science*, pages 18–30. Springer, 1997.
- [39] R.W. Brockett. Dynamical systems and their associated automata. In U. Helmke, R. Mennicken, and J. Saurer, editors, *Systems and Networks: Mathematical Theory and Applications MTNS 93*, volume 77, pages 49–69. Akademie-Verlag, Berlin, 1994.
- [40] M.C. Browne, E.M. Clarke, and O. Grumberg. Characterizing finite Kripke structures in propositional temporal logic. *Theoretical Computer Science*, 59:115–131, 1988.
- [41] J.R. Burch, E.M. Clarke, K.L. McMillan, D.L. Dill, and L.J. Hwang. Symbolic model checking: 10^{20} states and beyond. In *Proceedings of the Fifth Annual IEEE Symposium on Logic in Computer Science, LICS'90*, pages 1–33. IEEE Computer Society Press, 1990.
- [42] S.A. Bustin. Quantification of mRNA using real-time reverse transcription PCR (RT-PCR): Trends and problems. *Journal of Molecular Endocrinology*, 29(1):23–39, 2002.
- [43] L. Calzone, N. Chabrier-Rivier, F. Fages, L. Gentils, and S. Soliman. Machine learning bio-molecular interactions from temporal logic properties. In *Proceedings of the Third International Workshop on Computational Methods in Systems Biology, CMSB'05*, 2005.
- [44] R. Casey, H. de Jong, and J.-L. Gouzé. Piecewise-linear models of genetic regulatory networks: Equilibria and their stability. Technical Report RR-5353, INRIA Sophia-Antipolis, 2004.
- [45] R. Cavada, A. Cimatti, E. Olivetti, M. Pistore, and M. Roveri. *NuSMV 2.1 user manual*. ITC-IRST.
- [46] N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter. Modeling and querying biomolecular interaction networks. *Theoretical Computer Science*, 325(1):25–44, 2004.

- [47] C. Chaouiya, E. Remy, P. Ruet, and D. Thiéffry. Qualitative modelling of gene network: From logical regulatory graphs to standard Petri nets. In J. Cortadella and W. Reisig, editors, *Proceedings of the Twentyfifth International Conference on Applications and Theory of Petri Nets, ICATPN'04*, volume 3099 of *Lecture Notes in Computer Science*, pages 137–156. Springer, 2004.
- [48] A. Chutinan and B.H. Krogh. Verification of infinite-state dynamic systems using approximate quotient transition systems. *IEEE Transactions on Automatic Control*, 46(9):1401–1410, 2001.
- [49] A. Cimatti, E.M. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. NuSMV2: An opensource tool for symbolic model checking. In E. Brinksma and K.G. Larsen, editors, *Proceedings of the Fourteenth International Conference on Computer Aided Verification, CAV'02*, volume 2404 of *Lecture Notes in Computer Science*, pages 359–364. Springer, Copenhagen, Denmark, 2002.
- [50] A. Cimatti, E.M. Clarke, F. Giunchiglia, and M. Roveri. NuSMV: a new Symbolic Model Verifier. In N. Halbwachs and D. Peled, editors, *Proceedings of the Eleventh International Conference on Computer-Aided Verification, CAV'99*, number 1633 in *Lecture Notes in Computer Science*, pages 495–499, Trento, Italy, 1999. Springer-Verlag.
- [51] A. Cimatti, P.L. Pieraccini, R. Sebastiani, P. Traverso, and A. Villafiorita. Formal specification and validation of a vital communication protocol. In J.M. Wing, J. Woodcock, and J. Davies, editors, *Proceedings of the Second World Congress on Formal Methods, FM'99*, volume 1709 of *Lecture Notes in Computer Science*, pages 1584–1604. Springer, 1999.
- [52] E.M. Clarke, A. Biere, A. Cimatti, and Y. Zhu. Compositionality and abstraction - Symbolic model checking without BDDs. In W.R. Cleaveland, editor, *Proceedings of the Fifth International Conference on Tools and Algorithms for Construction and Analysis of Systems, TACAS'99*, volume 1579 of *Lecture Notes in Computer Science*, pages 193–207. Springer, 1999.
- [53] E.M. Clarke and I.A. Draghicescu. Expressibility results for linear-time and branching-time logics. In J.W. de Bakker, W.P. de Roever, and G. Rozenberg, editors, *REX Workshop*, volume 354 of *Lecture Notes in Computer Science*, pages 428–437. Springer, 1988.
- [54] E.M. Clarke and E.A. Emerson. Design and synthesis of synchronisation skeletons using branching-time temporal logic. In D. Kozen, editor, *Logic of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer, 1981.
- [55] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions On Programming Languages and Systems*, 8(2):244–263, 1986.
- [56] E.M. Clarke, A. Fehnker, Z. Han, B.H. Krogh, J. Ouaknine, O. Stursberg, and M. Theobald. Abstraction and counterexample-guided refinement in model checking of hybrid systems. *International Journal of Foundations of Computer Science*, 14(4):583–604, 2003.

- [57] E.M. Clarke, O. Grumberg, H. Hiraishi, S. Jha, D.E. Long, K.L. McMillan, and L.A. Ness. Verification of the futurebus+ cache coherence protocol. *Formal Methods in System Design*, 6(2):217–232, 1995.
- [58] E.M. Clarke, O. Grumberg, and D.E. Long. Model checking and abstraction. *ACM Transactions on Programming Languages and Systems*, 16(5):1512–42, 1994.
- [59] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. The MIT Press, Cambridge, MA, 1999.
- [60] R. Cleaveland and B. Steffen. A linear-time model-checking algorithm for the alternation-free modal mu-calculus. *Formal Methods in System Design*, 2:121–147, 1993.
- [61] J. Cohen. Classification of approaches used to study cell regulation: Search for a unified view using constraints. *Linköping Electronic Articles in Computer and Information Science*, 25(6), 2001.
- [62] J.R. Collier, N.A.M. Monk, P.K. Maini, and J.H. Lewis. Pattern formation by lateral inhibition with feedback: A mathematical model of Delta-Notch intercellular signalling. *Journal of Theoretical Biology*, 183(4):429–446, 1996.
- [63] O. Coudert, J.-C. Madre, and C. Berthet. Verifying temporal properties of sequential machines without building their state diagrams. In E.M. Clarke and R.P. Kurshan, editors, *Proceedings of the Second International Conference on Computer Aided Verification, CAV'90*, volume 531 of *Lecture Notes in Computer Science*, pages 23–32. Springer, 1990.
- [64] C. Daws, A. Olivero, S. Tripakis, and S. Yovine. The tool KRONOS. In R. Alur, T.A. Henzinger, and E.D. Sontag, editors, *Proceedings of the Third Workshop on Hybrid Systems: Verification and Control*, volume 1066 of *Lecture Notes in Computer Science*, pages 208–219. Springer, 1996.
- [65] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):69–105, 2002.
- [66] H. de Jong. Qualitative simulation and related approaches for the analysis of dynamic systems. *Knowledge Engineering Review*, 19(2):93–132, 2004.
- [67] H. de Jong, J. Geiselman, G. Batt, C. Hernandez, and M. Page. Qualitative simulation of the initiation of sporulation in *Bacillus subtilis*. *Bulletin of Mathematical Biology*, 66(2):261–299, 2004.
- [68] H. de Jong, J. Geiselman, C. Hernandez, and M. Page. Genetic Network Analyzer: Qualitative simulation of genetic regulatory networks. *Bioinformatics*, 19(3):336–344, 2003.
- [69] H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselman. Hybrid modeling and simulation of genetic regulatory networks: A qualitative approach. In A. Pnueli and O. Maler, editors, *Sixth International Workshop on Hybrid Systems: Computation and Control, HSCC'03*, volume 2623 of *Lecture Notes in Computer Science*, pages 267–282. Springer, 2003.

- [70] H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselman. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bulletin of Mathematical Biology*, 66(2):301–340, 2004.
- [71] P. Dembiński, W. Penczek, and A. Pórola. Verification of timed automata based on similarity. *Fundamenta Informaticae*, 50(1-2):59–89, 2002.
- [72] P.P. Dennis, M. Ehrenberg, and H. Bremer. Control of rRNA synthesis in *Escherichia coli*: A systems biology approach. *Microbiology and Molecular Biology Reviews*, 68(4):639–668, 2004.
- [73] L.G. Dixon, S. Sereidick, M. Richer, and G.B. Spiegelman. Developmental gene expression in *Bacillus subtilis* *crsA47* mutants reveals glucose-activated control of the gene for the minor sigma factor σ^H . *Journal of Bacteriology*, 183(16):4814–4822, 2001.
- [74] K. Drlica. Bacterial topoisomerases and the control of DNA supercoiling. *Trends in Genetics*, 6(12):433–437, 1990.
- [75] M.B. Dwyer, G.S. Avrunin, and J.C. Corbett. Property specification patterns for finite-state verification. In Mark Ardis, editor, *Proceedings of the Second Workshop on Formal Methods in Software Practice, FMSP '98*, pages 7–15. ACM Press, 1998.
- [76] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, and C.L. Talcott. Pathway logic: Executable models of biological networks. *Electronic Notes in Theoretical Computer Science*, 71, 2002.
- [77] H. El-Samad, S. Prajna, A. Papachristodoulou, M.H. Khammash, and J.C. Doyle. Model validation and robustness analysis of the bacterial heat shock response using SOSTOOLS. In *Proceedings of the Fortysecond IEEE Conference on Decision and Control, CDC'03*, 2003.
- [78] E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, pages 995–1072. MIT Press, 1990.
- [79] E.A. Emerson, editor. *Formal Methods in System Design*. Special Issue on Symmetry in Automatic Verification:9(1-2), 1996.
- [80] E.A. Emerson and J.Y. Halpern. ‘sometimes’ and ‘not never’ revisited: On branching versus linear time temporal logic. *Journal of the ACM*, 33(1):151–178, 1986.
- [81] D. Endy. Foundations for engineering biology. *Nature*, 438(7067):449–453, 2005.
- [82] S. Engell, S. Kowalewski, C. Schulz, and O. Stursberg. Continuous-discrete interactions in chemical processing plants. *Proceedings of the IEEE*, 88(7):1050–1066, 2000.
- [83] J. Errington. *Bacillus subtilis* sporulation: Regulation of gene expression and control of morphogenesis. *Microbiological Reviews*, 57(1):1–33, 1993.

- [84] D. Eveillard, D. Ropers, H. de Jong, C. Branlant, and A. Bockmayr. A multi-scale constraint programming model of alternative splicing regulation. *Theoretical Computer Science*, 325(1):3–24, 2004.
- [85] F. Fages, S. Soliman, and N. Chabrier-Rivier. Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. *Journal of Biological Physics and Chemistry*, 4(2):64–73, 2004.
- [86] E. Fanchon, F. Corblin, L. Trilling, B. Hermant, and D. Gulino. Modeling the molecular network controlling adhesion between human endothelial cells: Inference and simulation using constraint logic programming. In V. Danos and V. Schächter, editors, *Computational Methods in Systems Biology, CMSB'04*, volume 3082 of *Lecture Notes in Computer Science*, pages 104–118. Springer, 2005.
- [87] A.F. Filippov. *Differential Equations with Discontinuous Righthand Sides*. Kluwer Academic Publishers, 1988.
- [88] M. Fujita, J.E. Gonzalez-Pastor, and R. Losick. High- and low-threshold genes in the Spo0A regulon of *Bacillus subtilis*. *Journal of Bacteriology*, 187(4):1357–1368, 2005.
- [89] M. Fujita and R. Losick. Evidence that entry into sporulation in *Bacillus subtilis* is governed by a gradual increase in the level and activity of the master regulator Spo0A. *Genes and Development*, 19(18):2236–2244, 2005.
- [90] D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the temporal analysis of fairness. In *Proceedings of the Seventh Symposium on Principles of Programming Languages, POPL'80*, pages 163–173. ACM Press, 1980.
- [91] H. Garavel, F. Lang, and R. Mateescu. An overview of CADP 2001. Technical Report RR-0254, INRIA Rhône-Alpes, 2001.
- [92] H. Garavel and L. Mounier. Specification and verification of various distributed leader election algorithms for unidirectional ring networks. *Science of Computer Programming*, 29(1-2):171–198, 1997.
- [93] N.K. Gaur, K. Cabane, and I. Smith. Structure and expression of the *Bacillus subtilis* *sin* operon. *Journal of Bacteriology*, 170(3):1046–1053, 1988.
- [94] K. Gausing. Regulation of ribosome biosynthesis in *Escherichia coli*. In G. Chambliss, G.R. Graven, J. Davies, K. Davis, L. Kahan, and N. Nomura, editors, *Ribosomes: Structure, function, and genetics*, pages 693–718. University Park Press, Baltimore, 1980.
- [95] R. Ghosh, A. Tiwari, and C.J. Tomlin. Automated symbolic reachability analysis; with application to Delta-Notch signaling automata. In O. Maler and A. Pnueli, editors, *Sixth International Workshop on Hybrid Systems: Computation and Control, HSCC'03*, volume 2623 of *Lecture Notes in Computer Science*, pages 233–248. Springer, 2003.

- [96] R. Ghosh and C.J. Tomlin. Lateral inhibition through Delta-Notch signaling: A piecewise affine hybrid model. In M.D. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control, HSCC'01*, volume 2034 of *Lecture Notes in Computer Science*, pages 232–246. Springer, Berlin, 2001.
- [97] R. Ghosh and C.J. Tomlin. Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modelling: Delta-Notch protein signalling. *Systems Biology*, 1(1):170–183, 2004.
- [98] L. Glass. Combinatorial and topological methods in nonlinear chemical kinetics. *Journal of Chemical Physics*, 63(4):1325–1335, 1975.
- [99] L. Glass and S.A. Kauffman. The logical analysis of continuous non-linear biochemical control networks. *Journal of Theoretical Biology*, 39(1):103–129, 1973.
- [100] P. Godefroid. On the costs and benefits of using partial-order methods for the verification of concurrent systems. In D. Peled, V. Pratt, and G. Holzmann, editors, *Proceedings of DIMACS Workshop on Partial-Order Methods in Verification*, volume 29 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 289–304. AMS, 1997.
- [101] P. Godefroid and P. Wolper. A partial approach to model checking. *Information and Computation*, 110(2):305–326, 1994.
- [102] A. Goldbeter. *Biochemical oscillations and cellular rhythms: The molecular bases of periodic and chaotic behaviour*. Cambridge University Press, 1996.
- [103] A. Göllü and P. Varaiya. Hybrid dynamical systems. In *Proceedings of the Twentieth Conference on Decision and Control, CDC'89*, pages 2708–2712. IEEE Press, 1989.
- [104] G. Gonzalez-Gil, R. Kahmann, and G. Muskhelishvili. Regulation of *crp* transcription by oscillation between distinct nucleoprotein complexes. *EMBO Journal*, 17(10):2877–2885, 1998.
- [105] J.-L. Gouzé and T. Sari. A class of piecewise-linear differential equations arising in biological models. *Dynamical Systems*, 17(4):299–316, 2002.
- [106] S. Graf and H. Saidi. Construction of abstract state graphs with PVS. In *Proceedings of the Ninth International Conference on Computer Aided Verification, CAV'97*, volume 1254 of *Lecture Notes in Computer Science*, pages 72–83. Springer, 1997.
- [107] A.D. Grossman. Genetic networks controlling the initiation of sporulation and the development of genetic competence in *Bacillus subtilis*. *Annual Review of Genetics*, 29:477–508, 1995.
- [108] O. Grumberg and D.E. Long. Model checking and modular verification. *ACM Transactions on Programming Languages and Systems*, 16(3):1994, 1994.
- [109] W. Hamscher, L. Console, and J. de Kleer, editors. *Readings in Model-Based Diagnosis*. Morgan Kaufmann, 1992.

- [110] V. Hartonas-Garmhausen, S.V.A. Campos, A. Cimatti, E.M. Clarke, and F. Giunchiglia. Verification of a safety-critical railway interlocking system with real-time constraints. *Science of Computer Programming*, 36(1):53–64, 2000.
- [111] L.H. Hartwell, J.J. Hopfield, S. Leibler, and A.W. Murray. From molecular to modular cell biology. *Nature*, 402(supplement):C47–C52, 1999.
- [112] J. Hasty, D. McMillen, F. Isaacs, and J.J. Collins. Computational studies of gene regulatory networks: *in numero* molecular biology. *Nature Review Genetics*, 2(4):268–279, 2001.
- [113] K. Havelund, M.R. Lowry, and J. Penix. Formal analysis of a space-craft controller using SPIN. *IEEE Transactions on Software Engineering*, 27(8):749–765, 2001.
- [114] M. Hecker and U. Völker. General stress response of *Bacillus subtilis* and other bacteria. *Advances in Microbial Physiology*, 44:35–91, 2001.
- [115] K.R. Heidtke and S. Schulze-Kremer. Design and implementation of a qualitative simulation model of lambda phage infection. *Bioinformatics*, 14(1):81–91, 1998.
- [116] M. Heiner, I. Koch, and J. Will. Model validation of biological pathways using Petri nets - demonstrated for apoptosis. *BioSystems*, 75:15–28, 2004.
- [117] R. Hengge-Aronis. The general stress response in *Escherichia coli*. In G. Storz and R. Hengge-Aronis, editors, *Bacterial stress responses*, pages 161–177. ASM Press, 2000.
- [118] T.A. Henzinger. Hybrid automata with finite bisimulations. In Z. Fülöp and F. Gécseg, editors, *Proceedings of the Twentysecond International Colloquium on Automata, Languages and Programming, ICALP'95*, volume 944 of *Lecture Notes in Computer Science*, pages 324–335. Springer, 1995.
- [119] T.A. Henzinger, P. Ho, and H. Wong-Toi. HYTECH: A model checker for hybrid systems. *International Journal on Software Tools for Technology Transfer*, 1(1-2):110–122, 1997.
- [120] T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? *Journal of Computer and System Sciences*, 57:94–124, 1998.
- [121] V.J. Hernandez and H. Bremer. Guanosine tetraphosphate (ppgpp) dependence of the growth rate control of *rrnB* P1 promoter activity in *Escherichia coli*. *Journal of Biological Chemistry*, 265(20):11605–11614, 1990.
- [122] J.A. Hoch. Genetics of bacterial sporulation. *Advances in Genetics*, 18:69–98, 1976.
- [123] J.A. Hoch. *spo0* genes, the phosphorelay, and the initiation of sporulation. In A.L. Sonenshein, J.A. Hoch, and R. Losick, editors, *Bacillus subtilis and other Gram-Positive Bacteria: Biochemistry, Physiology, and Molecular Genetics*, pages 747–756. ASM Press, 1993.
- [124] R. Horowitz and P. Varaiya. Control design of an automated highway system. *Proceedings of the IEEE*, 88(7):913–925, 2000.

- [125] G.J. Hotzmann. The model checker SPIN. *IEEE Transactions on Software Engineering*, 23(5):279–295, 1997.
- [126] J. Hu, W.-C. Wu, and S. Sastry. Modeling subtilin production in *Bacillus subtilis* using stochastic hybrid systems. In R. Alur and G.J. Pappas, editors, *Seventh International Workshop on Hybrid Systems: Computation and Control, HSCC'04*, volume 2993 of *Lecture Notes in Computer Science*, pages 417–431. Springer, 2004.
- [127] G.W. Huisman, D.A. Siegele, M.M. Zambrano, and R. Kolter. Morphological and physiological changes during stationary phase. In F.C. Neidhardt, R. Curtiss III, J.L. Ingraham, E.C.C. Lin, K.B. Low, B. Magasanik, W.S. Reznikoff, M. Riley, M. Schaechter, and H.E. Umbarger, editors, *Escherichia coli and Salmonella: Cellular and Molecular Biology*, pages 1672–1682. ASM Press, 1996.
- [128] T. Hunkapiller, R.J. Kaiser, B.F. Koop, and L.E. Hood. Large-scale and automated DNA sequence determination. *Science*, 254(5028):59–67, 1991.
- [129] H. Ishizuka, A. Hanamura, T. Inada, and H. Aiba. Mechanism of the down-regulation of cAMP receptor protein by glucose in *Escherichia coli*: Role of autoregulation of the *crp* gene. *EMBO Journal*, 13(13):3077–82, 1994.
- [130] M. Jiang, W. Shao, M. Perego, and J.A. Hoch. Multiple histidine kinases regulate entry into stationary phase and sporulation in *Bacillus subtilis*. *Molecular Microbiology*, 38(3):535–542, 2000.
- [131] S.A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467, 1969.
- [132] I.M. Keseler, J. Collado-Vides, S. Gama-Castro, J. Ingraham, S. Paley, I.T. Paulsen, M. Peralta-Gil, and P.D. Karp. EcoCyc: A comprehensive database resource for *Escherichia coli*. *Nucleic Acids Research*, 33:D334–D337, 2005.
- [133] P.M. Kim and B. Tidor. Limitations of quantitative gene regulation models: A case study. *Genome Research*, 13(11):2391–2395, 2003.
- [134] R.D. King, S.M. Garrett, and G.M. Coghill. On the use of qualitative reasoning to simulate and identify metabolic pathways. *Bioinformatics*, 21(9):2017–2026, 2005.
- [135] H. Kitano. Systems biology: A brief overview. *Science*, 295(5560):1662–1664, 2002.
- [136] I. Koch, B.H. Junker, and M. Heiner. Application of Petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. *Bioinformatics*, 21(7):1219–1226, 2005.
- [137] K.W. Kohn. Molecular interaction maps as information organizers and simulation guides. *Chaos*, 11(1):1–14, 2001.
- [138] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.
- [139] B. Kuipers and S. Ramamoorthy. Qualitative modeling and heterogeneous control of global system behavior. In C.J. Tomlin and M.R. Greenstreet, editors, *Hybrid Systems: Computation and Control, HSCC'02*, volume 2289 of *Lecture Notes in Computer Science*, pages 294–307. Springer, 2002.

- [140] B.J. Kuipers. Qualitative simulation. *Artificial Intelligence*, 29(3):289–388, 1986.
- [141] B.J. Kuipers. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT Press, 1994.
- [142] O. Kupferman and M.Y. Vardi. Modular model checking. In *Proceedings of Compositionality - The Significant Difference*, volume 1536 of *Lecture Notes in Computer Science*. Springer, 1998.
- [143] O. Kupferman, M.Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–60, 2000.
- [144] T.I. Lee, N.J. Rinaldi, F. Robert, D.T. Odom, Z. Bar-Joseph, G.K. Gerber, N.M. Hannett, C.T. Harbison, C.M. Thompson, I. Simon, J. Zeitlinger, E.G. Jennings, H.L. Murray, D.B. Gordon, B. Ren, J.J. Wyrick, J.-B. Tagne, T.L. Volkert, E. Fraenkel, D.K. Gifford, and R.A. Young. Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, 298(5594):799–804, 2002.
- [145] B. Lennartson, M. Tittus, B. Egardt, and S. Pettersson. Hybrid systems in process control. *IEEE Control Systems Magazine*, 16(5):45–56, 1996.
- [146] B. Lewin. *Genes VIII*. Prentice Hall, 2004.
- [147] C. Livadas, J. Lygeros, and N.A. Lynch. High-level modeling and analysis of traffic alert and avoidance system. *Proceedings of the IEEE*, 88(7):926–948, 2000.
- [148] D.J. Lockhart and E.A. Winzeler. Genomics, gene expression and DNA arrays. *Nature*, 405(6788):827–836, 2000.
- [149] C. Loiseaux, S. Graf, J. Sifakis, A. Bouajjani, and S. Bensalem. Property preserving abstractions for the verification of concurrent systems. *Formal Methods in System Design*, 6(1), 1994.
- [150] J. Lygeros. Lecture notes on hybrid systems, 2004.
- [151] J. Lygeros, D.N. Godbole, and S. Sastry. Verified hybrid controllers for automated vehicles. *IEEE Transactions on Automatic Control*, 43(4):522–539, 1998.
- [152] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3):349–370, 1999.
- [153] O. Maler, Z. Manna, and A. Pnueli. From timed to hybrid systems. In J.W. de Bakker, C. Huizing, W.P. de Roever, and G. Rozenberg, editors, *Proceedings of the REX Workshop on Real-Time: Theory in Practice*, volume 600 of *Lecture Notes in Computer Science*, pages 447–484. Springer, 1991.
- [154] I. Mandić-Mulec, L. Doukhan, and I. Smith. The *Bacillus subtilis* SinR protein is a repressor of the key sporulation gene *spo0A*. *Journal of Bacteriology*, 177(16):4619–4627, 1995.
- [155] G. Marnellos, G.A. Deblandre, E. Mjolsness, and C. Kintner. Delta-notch lateral inhibitory patterning in the emergence of ciliated cells in xenopus: Experimental observations and a gene network model. In *Proceedings of the Pacific Symposium on Biocomputing, PSB'00*, volume 5, pages 326–337. World Scientific Publishing, 2000.

- [156] A. Martinez-Antonio and J. Collado-Vides. Identifying global regulators in transcriptional regulatory networks in bacteria. *Current Opinion in Microbiology*, 6(5):482–489, 2003.
- [157] R. Mateescu. Notes on verifying Kripke structure using action-based logics. Contact: Radu.Mateescu@inria.fr, 2005.
- [158] R. Mateescu and H. Garavel. XTL: A meta-language and tool for temporal logic model-checking. In *Proceedings of the International Workshop on Software Tools for Technology Transfer, STTT'98*, pages 33–42, 1998.
- [159] R. Mateescu and M. Sighireanu. Efficient on-the-fly model-checking for regular alternation-free mu-calculus. *Science of Computer Programming*, 46(3):255–281, 2003.
- [160] H.H. McAdams and A. Arkin. Simulation of prokaryotic genetic circuits. *Annual Review of Biophysics and Biomolecular Structure*, 27:199–224, 1998.
- [161] H.H. McAdams and L. Shapiro. Circuit simulation of genetic networks. *Science*, 269(5224):650–656, 1995.
- [162] S. Merz. Model checking: A tutorial overview. In F. Cassez, C. Jard, B. Rozoy, and M. D. Ryan, editors, *Lecture Notes of the Fourth Summer School on Modeling and Verification of Parallel Processes, MOVEP'00*, volume 2067 of *Lecture Notes in Computer Science*, pages 3–38. Springer, 2001.
- [163] T. Mestl, E. Plahte, and S.W. Omholt. A mathematical framework for describing and analysing gene regulatory networks. *Journal of Theoretical Biology*, 176:291–300, 1995.
- [164] R. Milner. An algebraic definition of simulation between programs. In D.C. Cooper, editor, *Proceedings of the Second International Joint Conference on Artificial Intelligence, IJCAI-71*, pages 481–489. Morgan Kaufmann, 1971.
- [165] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer, 1980.
- [166] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.
- [167] M.O. Möller, H. Rueß, and M. Sorea. Predicate abstraction for dense real-time systems. *Electronic Notes in Theoretical Computer Science*, 65(6), 2002.
- [168] J. Monod and F. Jacob. General conclusions: Teleonomic mechanisms in cellular metabolism, growth, and differentiation. In *Cold Spring Harbor Symposium on Quantitative Biology*, volume 26, pages 389–401, 1961.
- [169] I. Moszer, L.M. Jones, S. Moreira, C. Fabry, and A. Danchin. SubtiList: The reference database for the *Bacillus subtilis* genome. *Nucleic Acids Research*, 30(1):62–65, 2002.
- [170] A. Nerode and W. Kohn. Models for hybrid systems: Automata, topologies, controllability, observability. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *First International Workshop on Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 317–356. Springer, 1993.

- [171] X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. An approach to the description and analysis of hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *First International Workshop on Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 149–178. Springer, 1993.
- [172] X. Nicollin, J. Sifakis, and S. Yovine. From ATP to timed graphs and hybrid systems. *Acta Informatica*, 30(2):181–202, 1993.
- [173] L. Nilsson, H. Verbeek, E. Vijgenboom, C. van Drunen, A. Vanet, and L. Bosch. FIS-dependent trans activation of stable RNA operons of *Escherichia coli* under various growth conditions. *Journal of Bacteriology*, 174(3):921–929, 1992.
- [174] T.J. Oh, I.L. Jung, and I.G. Kim. The *Escherichia coli* SOS gene *sbmC* (*gyrI*) is regulated by H-NS and RpoS during the SOS induction and stationary growth phase. *Biochemical and Biophysical Research Communications*, 288(4):1052–1058, 2001.
- [175] A. Pandey and M. Mann. Proteomics to study genes and genomes. *Nature*, 405(6788):837–846, 2000.
- [176] D. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *fifth GI-Conference on Theoretical Computer Science*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer, 1981.
- [177] D.L. Pepyne and C.G. Cassandras. Optimal control of hybrid systems in manufacturing. *Proceedings of the IEEE*, 88(7):1108–1122, 2000.
- [178] M. Perego. Kinase-phosphatase competition regulates *Bacillus subtilis* development. *Trends in Microbiology*, 6(9):366–370, 1998.
- [179] M. Perego and J.A. Hoch. Sequence analysis of the *hpr* locus, a regulatory gene for protease production and sporulation in *Bacillus subtilis*. *Journal of Bacteriology*, 170(6):2560–2567, 1988.
- [180] M. Perego, G.B. Spiegelman, and J.A. Hoch. Structure of the gene for the transition state regulator, *abrB*: Regulator synthesis is controlled by the *spo0A* sporulation gene in *Bacillus subtilis*. *Molecular Microbiology*, 2(6):689–699, 1988.
- [181] B.J. Peter, J. Arsuaga, A.M. Breier, A.B. Khodursky, P.O. Brown, and N.R. Cozzarelli. Genomic transcriptional response to loss of chromosomal supercoiling in *Escherichia coli*. *Genome Biology*, 5(11):R87, 2004.
- [182] N.B.O.L. Pettit. *Analysis of Piecewise Linear Dynamical Systems*. Number 3 in UMIST Control Systems Centre Series. Research Studies Press, 1999.
- [183] R.D. Phair and T. Misteli. Kinetic modelling approaches to *in vivo* imaging. *Nature Reviews Molecular Cell Biology*, 2(12):898–907, 2001.
- [184] E. Plahte, T. Mestl, and S.W. Omholt. Global analysis of steady points for systems of differential equations with sigmoid interactions. *Dynamics and Stability of Systems*, 9(4):275–291, 1994.

- [185] T.S. Pratt, T. Steiner, L.S. Feldman, K.A. Walker, and R. Osuna. Deletion analysis of the *fis* promoter region in *Escherichia coli*: Antagonistic effects of Integration Host Factor and Fis. *Journal of Bacteriology*, 179(20):6367–6377, 1997.
- [186] M. Ptashne. *A Genetic Switch: Phage λ and Higher Organisms*. Cell Press and Blackwell Science, Cambridge, MA, 2nd edition, 1992.
- [187] A. Puri and P. Varaiya. Driving safely in smart cars. Technical Report UCB-ITS-PRR-95-24, University of California, Berkely, 1995.
- [188] J.-P. Queille and J. Sifakis. Specification and verification of concurrent systems in CESAR. In M. Dezani-Ciancaglini and U. Montanari, editors, *Proceedings of the Fifth Colloquium on International Symposium on Programming*, volume 137 of *Lecture Notes in Computer Science*, pages 337–351. Springer, 1982.
- [189] V.N. Reddy, M.N. Liebman, and M.L. Mavrovouniotis. Qualitative analysis of biochemical reaction systems. *Computers in biology and medicine*, 26(1):9–24, 1996.
- [190] A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using the π -calculus process algebra. In R.B. Altman, A.K. Dunker, L. Hunter, K. Lauderdale, and T.E. Klein, editors, *Pacific Symposium on Biocomputing, PSB'01*, volume 6, pages 459–470. World Scientific Publishing, 2001.
- [191] A. Richard, J.-P. Comet, and G. Bernot. R. Thomas' modeling of biological regulatory networks: Introduction of singular states in the qualitative dynamics. *Fundamenta Informaticae*, 65(4):373–392, 2005.
- [192] D. Ropers, H. de Jong, M. Page, D. Schneider, and J. Geiselmann. Qualitative simulation of the carbon starvation response in *Escherichia coli*. *BioSystems*, 2005. Epub ahead of print: <http://dx.doi.org/10.1016/j.biosystems.2005.10.005>.
- [193] E.P. Sacks and J. Doyle. Prolegomena to any future qualitative physics. *Computational Intelligence*, 8(2):187–209, 1992.
- [194] M.H. Saier, T.M. Ramseier, and J. Reizer. Regulation of carbon utilization. In F.C. Neidhardt, R. Curtiss III, J.L. Ingraham, E.C.C. Lin, K.B. Low, B. Magasanik, W.S. Reznikoff, M. Riley, M. Schaechter, and H.E. Umbarger, editors, *Escherichia coli and Salmonella: Cellular and Molecular Biology*, pages 1325–1343. ASM Press, 1996.
- [195] H. Salgado, S. Gama-Castro, A. Martinez-Antonio, E. Diaz-Peredo, F. Sanchez-Solano, M. Peralta-Gil, D. Garcia-Alonso, V. Jimenez-Jacinto, A. Santos-Zavaleta, C. Bonavides-Martinez, and J. Collado-Vides. RegulonDB (version 4.0): Transcriptional regulation, operon organization and growth conditions in *Escherichia coli* K-12. *Nucleic Acids Research*, 1(32):D303–D306, 2004.
- [196] P. Schaeffer, J. Millet, and J.P. Aubert. Catabolic repression of bacterial sporulation. *Proceedings of the National Academy of Sciences of the USA*, 54(3):704–711, 1965.
- [197] S.H. Shafikhani, I. Mandic-Mulec, M.A. Strauch, I. Smith, and T. Leighton. Post-exponential regulation of *sin* operon expression in *Bacillus subtilis*. *Journal of Bacteriology*, 184(2):564–571, 2002.

- [198] A. Siegel, O. Radulescu, M. Le Borgne, P. Veber, J. Ouy, and S. Lagarrigue. Qualitative analysis of the relation between DNA microarray data and behavioral models of regulation networks. *BioSystems*, 2005. To appear.
- [199] P. Smolen, D.A. Baxter, and J.H. Byrne. Modeling transcriptional control in gene networks: Methods, recent results, and future directions. *Bulletin of Mathematical Biology*, 62(2):247–292, 2000.
- [200] E.H. Snoussi. Qualitative dynamics of piecewise-linear differential equations: A discrete mapping approach. *Dynamics and Stability of Systems*, 4(3-4):189–207, 1989.
- [201] E.H. Snoussi and R. Thomas. Logical identification of all steady states: The concept of feedback loop characteristic states. *Bulletin of Mathematical Biology*, 55(5):973–991, 1993.
- [202] O. Sokolsky and H.S. Hong. Qualitative modeling of hybrid systems. In *Proceedings of the Monterey Workshop on Engineering Automation for Computer Based Systems*, 2001.
- [203] A.L. Sonenshein. Bacterial sporulation: A response to environmental signals. In G. Storz and R. Hengge-Aronis, editors, *Bacterial Stress Responses*, chapter 13, pages 199–215. ASM Press, 2000.
- [204] M. Song, T.-J. Tarn, and N. Xi. Integration of task scheduling, action planning, and control in robotic manufacturing systems. *Proceedings of the IEEE*, 88(7):1097–1107, 2000.
- [205] K. Stephenson and R.J. Lewis. Molecular insights into the initiation of sporulation in Gram-positive bacteria: New technologies for an old phenomenon. *FEMS Microbiology Review*, 29(2):281–301, 2005.
- [206] P. Stragier and R. Losick. Molecular genetics of sporulation in *Bacillus subtilis*. *Annual Review of Genetics*, 30:297–341, 1996.
- [207] M.A. Strauch. AbrB, a transition state regulator. In A.L. Sonenshein, J.A. Hoch, and R. Losick, editors, *Bacillus subtilis and other Gram-Positive Bacteria: Biochemistry, Physiology, and Molecular Genetics*, pages 757–764. ASM Press, 1993.
- [208] M.A. Strauch and J.A. Hoch. Transition-state regulators: Sentinels of *Bacillus subtilis* post-exponential gene expression. *Molecular Microbiology*, 7(3):337–342, 1993.
- [209] M. Sugita. Functional analysis of chemical systems *in vivo* using a logical circuit equivalent. *Journal of Theoretical Biology*, 1:179–192, 1961.
- [210] C.L. Talcott, S. Eker, M. Knapp, P. Lincoln, and K. Laderoute. Pathway logic modeling of protein functional domains in signal transduction. In R.B. Altman, A. Keith Dunker, L. Hunter, T.A. Jung, and T.E. Klein, editors, *Pacific Symposium on Biocomputing, PSB'04*, pages 568–580. World Scientific Publishing, 2004.
- [211] L. Tavernini. Differential automata and their discrete simulators. *Nonlinear Analysis, Theory, Methods and Applications*, 11(6):665–683, 1987.

- [212] R. Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42:563–585, 1973.
- [213] R. Thomas. Regulatory networks seen as asynchronous automata: A logical description. *Journal of Theoretical Biology*, 153:1–23, 1991.
- [214] R. Thomas and R. d’Ari. *Biological Feedback*. CRC Press, 1990.
- [215] R. Thomas, D. Thieffry, and M. Kaufman. Dynamical behaviour of biological regulatory networks: I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of Mathematical Biology*, 57(2):247–276, 1995.
- [216] A. Tiwari and G. Khanna. Series of abstractions for hybrid automata. In C.J. Tomlin and M.R. Greenstreet, editors, *Hybrid Systems: Computation and Control HSCC’02*, volume 2289 of *Lecture Notes in Computer Science*, pages 465–478. Springer, 2002.
- [217] C.J. Tomlin, G.J. Pappas, and S. Sastry. Conflict resolution for air traffic management: A study in multi-agent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, 1998.
- [218] L. Travé-Massuyès, L. Ironi, and P. Dague. Mathematical foundations of qualitative reasoning. *AI Magazine*, 24(4):91–106, 2003.
- [219] A. Travers, R. Schneider, and G. Muskhelishvili. DNA supercoiling and transcription in *Escherichia coli*: the Fis connection. *Biochimie*, 83(2):213–217, 2001.
- [220] S. Tripakis and S. Yovine. Analysis of timed systems using time-abstracting bisimulations. *Formal Methods in System Design*, 18(1):25–68, 2001.
- [221] P. Uetz, L. Giot, G. Cagney, T.A. Mansfield, R.S. Judson, J.R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M.J. Yang, M. Johnston, S. Fields, and J.M. Rothberg. A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. *Nature*, 403(6770):623–627, 2000.
- [222] A. Usseglio-Viretta and M. Fussenegger. Modeling the quorum sensing regulatory network of human-pathogenic *Pseudomonas aeruginosa*. *Biotechnology Progress*, 20(3):670–678, 2004.
- [223] J. Weir, M. Predich, E. Dubnau, G. Nair, and I. Smith. Regulation of *spo0H*, a gene coding for the *Bacillus subtilis* σ^H factor. *Journal of Bacteriology*, 173(2):521–529, 1991.
- [224] H.V. Westerhoff and B.O. Palsson. The evolution of molecular biology into systems biology. *Nature Biotechnology*, 22(10):1249–1252, 2004.
- [225] L.M. Wick and T. Egli. Molecular components of physiological stress responses in *Escherichia coli*. *Advances in Biochemical Engineering/Biotechnology*, 89:1–45, 2004.

- [226] S. Yamashita, F. Kawamura, H. Yoshikawa, H. Takahashi, Y. Kobayashi, and H. Saito. Dissection of the expression signals of the *spo0A* gene of *Bacillus subtilis*: Glucose represses sporulation-specific expression. *Journal of General Microbiology*, 135(5):1335–1345, 1989.
- [227] T.-M. Yi, M. Fazel, X. Liu, O. Otitoju, J. Goncalves, A. Papachristodoulou, S. Prajna, and J. C. Doyle. Application of robust model validation using SOSTOOLS to the study of G-protein signaling in yeast. In *Proceedings of the First Conference on Foundations of Systems Biology in Engineering, FOSBE'05*, 2005.
- [228] I. Zevedei-Oancea and S. Schuster. Topological analysis of metabolic networks based on Petri-net theory. *In Silico Biology*, 3:29, 2003.

Résumé

Les réseaux de régulation génique contrôlent le développement et le fonctionnement des organismes vivants. Etant donné que la plupart des réseaux de régulation génique d'intérêt biologique sont grands et que leur dynamique est complexe, la compréhension de leur fonctionnement est un problème biologique majeur. De nombreuses méthodes ont été développées pour la modélisation et la simulation de ces systèmes. Étonnamment, le problème de la validation de modèle n'a reçu jusqu'à récemment que peu d'attention. Pourtant, cette étape est d'autant plus importante que dans le contexte de la modélisation de réseaux de régulation génique, les systèmes modélisés sont complexes et encore imparfaitement connus.

Dans cette thèse, nous proposons une approche permettant de tester la validité de modèles de réseaux de régulation génique en comparant les prédictions obtenues avec les données expérimentales. Plus spécifiquement, nous considérons dans ce travail une classe de modèles qualitatifs définis en termes d'équations différentielles linéaires par morceaux (LPM). Ces modèles permettent de capturer les aspects essentiels des régulations géniques, tout en ayant une forme mathématique simple qui facilite leur analyse symbolique. Également, nous souhaitons utiliser les informations qualitatives sur la dynamique du système données par les changements du sens de variation des concentrations des protéines du réseau. Ces informations peuvent être obtenues expérimentalement à partir de profils d'expression temporels.

La méthode proposée doit satisfaire deux contraintes. Premièrement, elle doit permettre d'obtenir des prédictions bien adaptées à la comparaison avec le type de données considéré. Deuxièmement, étant donné la taille et la complexité des réseaux d'intérêt biologique, la méthode doit également permettre de vérifier efficacement la cohérence entre prédictions et observations.

Pour répondre à ces deux contraintes, nous étendons dans deux directions une approche précédemment développée par de Jong et collègues pour l'analyse symbolique des modèles LPM qualitatifs. Premièrement, nous proposons d'utiliser une représentation plus fine de l'état du système, permettant d'obtenir, par abstraction discrète, des prédictions mieux adaptées à la comparaison avec les données expérimentales. Deuxièmement, nous proposons de combiner cette méthode avec des techniques de model checking. Nous montrons que l'utilisation combinée d'abstraction discrète et de model checking permet de vérifier efficacement les propriétés dynamiques, exprimées en logique temporelle, des modèles continus.

Cette méthode a été implémentée dans une nouvelle version de l'outil Genetic Network Analyzer (GNA 6.0). GNA 6.0 a été utilisé pour la validation de deux modèles grands et complexes de l'initiation de la sporulation chez *B. subtilis* et de la réponse au stress nutritionnel chez *E. coli*. Nous avons ainsi pu vérifier que les prédictions obtenues étaient en accord avec la plupart des données expérimentales disponibles dans la littérature. Plusieurs incohérences ont également été identifiées, suggérant des révisions des modèles ou la réalisation d'expériences complémentaires. En dehors d'une contribution à une meilleure compréhension du fonctionnement de ces systèmes, ces deux études de cas illustrent plus généralement que, par la méthode proposée, il est possible de tester si des prédictions obtenues pour des modèles complexes sont cohérentes avec un large éventail de propriétés observables expérimentalement.

Abstract

Genetic regulatory networks control the development and functioning of living organisms. Given that most of the genetic regulatory networks of biological interest are large and that their dynamics are complex, understanding their functioning is a major biological problem. Numerous methods have been developed for the modeling and simulation of such systems. Surprisingly, the problem of model validation has received little attention until recently. However, the validation step is very important since the modeled systems are complex and still imperfectly known.

In this thesis, we propose an approach for testing the validity of models of genetic regulatory networks by comparing the predictions obtained from the model with experimental data. We consider a class of qualitative models of genetic regulatory networks defined in terms of piecewise-linear (PL) differential equations. While having a simple mathematical form that favors their symbolic analysis, these models capture essential aspects of genetic regulation. As experimental information on the dynamics of the system, we consider the time evolution of the direction of change of protein concentrations in the network. This information can be experimentally obtained from temporal expression profiles.

The method that we propose must satisfy two constraints. First, it should provide predictions that are well-adapted to comparison with the type of data we consider. Second, it should allow for efficient checks of consistency between predictions and observations, given the size and the complexity of the networks of biological interest.

To meet these two constraints, we extend an approach developed by de Jong and colleagues for symbolic analysis of qualitative PL models in two directions. First, we propose to use a finer representation of the state of the system. Through the use of a discrete abstraction, this allows us to obtain predictions that are better adapted to comparison with experimental data. Second, we propose to combine this method with model-checking techniques. We demonstrate that the combined use of discrete abstraction and model checking makes it possible to check efficiently dynamical properties, expressed in temporal logic, of continuous models.

This method has been implemented in a new version of the tool Genetic Network Analyzer (GNA 6.0). GNA 6.0 has been used for the validation of two complex models. The first is a model of the initiation of sporulation in *B. subtilis*. The second is a model of the nutritional stress response in *E. coli*. For both systems, we verify that the predictions obtained from the models are consistent with most of the experimental data available in the literature. Several inconsistencies have also been identified, suggesting either model revisions or the realization of complementary experiments. In addition to contributing to a better understanding of the functioning of these systems, these two case studies illustrate more generally that using the method we propose it is possible to test whether the predictions obtained from complex models are consistent with a variety of experimentally-observed properties.