



HAL
open science

Conception et mise en oeuvre d'un outil déclaratif pour l'analyse des réseaux génétiques discrets

Fabien Corblin

► **To cite this version:**

Fabien Corblin. Conception et mise en oeuvre d'un outil déclaratif pour l'analyse des réseaux génétiques discrets. Informatique [cs]. Université Joseph-Fourier - Grenoble I, 2008. Français. NNT : . tel-00388776

HAL Id: tel-00388776

<https://theses.hal.science/tel-00388776>

Submitted on 27 May 2009

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.

Conception et mise en œuvre d'un outil déclaratif pour l'analyse des réseaux génétiques discrets.

THÈSE

présentée et soutenue publiquement le 8 décembre 2008

pour l'obtention du

Doctorat de l'université Joseph Fourier – Grenoble 1
(spécialité informatique)

par

Fabien Corblin

Composition du jury

Président : Jacques Demongeot

Rapporteurs : Gilles Bernot
Denis Thieffry

Examineur : Youssef Hamadi

Co-Directeur : Eric Fanchon

Directeur : Laurent Trilling

Mis en page avec la classe thloria.

Remerciements

Mes premiers remerciements vont à mes encadrants, Eric Fanchon et Laurent Trilling, pour avoir suivi mon travail et m'avoir guidé avec tant d'énergie et de constance.

Ensuite je tiens à remercier mes encadrants à l'étranger, Jacques Cohen et Youssef Hamadi, pour m'avoir accueilli et s'être plongés dans mon sujet de thèse avec tant d'enthousiasme, ce qui a conduit à ouvrir nos recherches de manière très constructive.

Je remercie aussi Microsoft Research pour avoir financé mes trois années de thèse.

Je remercie également nos autres collaborateurs Lucas Bordeaux, Claudine Chaouiya, Delphine Ropers et Denis Thieffry pour leur générosité dans ce travail collaboratif.

Je remercie mes amis et collègues des quatre laboratoires où j'ai pu travailler durant la préparation de la thèse : TIMC-IMAG (Grenoble), LSR-IMAG (Grenoble), Volen Center for Complex Systems (US), Microsoft Research Cambridge (UK), pour leur soutien et nos échanges passionnants qui ont conduit à clarifier mes travaux.

Merci enfin à mes parents, mes deux sœurs, mon maître d'Aïkido, Carole, Mathias, Xavier, et tous mes autres amis qui se reconnaîtront pour m'avoir conseillé, encouragé et surtout ouvert les yeux sur certains aspects essentiels de mon travail d'enseignant-chercheur.

Huit forces soutiennent la Création :
Le mouvement et l'immobilité
La solidification et la fluidité
L'extension et la contraction
L'unification et la division

Morihei Ueshiba, fondateur de l'Aïkido

Résumé

Une demande croissante d'outils pour construire et décrypter des réseaux génétiques contrôlant des processus cellulaires est ressentie en biologie. Nous soutenons que l'utilisation de l'approche déclarative est pertinente et applicable pour répondre aux questions des biologistes sur ces réseaux, en général partiellement connus. L'idée principale est de modéliser des connaissances portant à la fois sur la structure et la dynamique d'un réseau par un ensemble de contraintes représentant l'ensemble des solutions, de vérifier sa cohérence, de réparer une incohérence éventuelle par un relâchement automatique, et d'inférer des propriétés sur la structure et la dynamique du réseau. Pour montrer la faisabilité de l'approche, nous formalisons les réseaux discrets de R. Thomas et les propriétés biologiques pertinentes, proposons un outil reposant sur la Programmation Logique par Contraintes en coopération avec un solveur SAT, et la validons sur des applications biologiques significatives.

Mots-clés: Bioinformatique, Réseaux génétiques, Intelligence Artificielle, Programmation Logique par Contraintes, Satisfiabilité Booléenne, Stress nutritionnel, Contrôle de l'immunité, Segmentation de l'embryon.

Abstract

A growing demand for tools to build and decrypt genetic networks that control cellular processes is felt in biology. We argue that the use of the declarative approach is relevant and applicable to answer questions from biologists about these networks, which are in general partially known. The main idea is to model knowledge about both the structure and the dynamic of a network by a set of constraints representing all the solutions, to check its consistency, to repair a possible inconsistency by an automatic constraint removal, and to infer properties on the structure and dynamic of the network. To demonstrate the feasibility of the approach, we formalize the discrete networks of R. Thomas and relevant biological properties, offer a tool based on constraint logic programming in cooperation with a SAT solver, and validate it on significant biological applications.

Keywords: Bioinformatics, Genetic Networks, Artificial Intelligence, Constraint Logic Programming, Boolean Satisfiability, Nutritional stress, Immunity control, Embryo segmentation.

Title: Design and implementation of a declarative tool for analyzing discrete genetic networks.

Table des matières

Introduction

Chapitre 1

Introduction

1.1	Cadre	3
1.2	Sujet	3
1.2.1	Problématique	4
1.2.2	Approche informatique	5
1.2.3	Illustration du problème et de l'approche à l'aide d'un exemple	6
1.3	Plan	7

Chapitre 2

Travaux connexes

2.1	Analyse de réseaux dont les paramètres sont fixés	9
2.2	Analyse de réseaux dont les paramètres ne sont pas fixés	10
2.3	Comparaison avec notre proposition	11

Formalisation du problème

Chapitre 3

Formalisation des réseaux de Thomas en contraintes

3.1	Définition d'un modèle \mathcal{M}	15
3.1.1	Notion d'interaction et d'activité d'une interaction	16
3.1.2	Notion de contexte cellulaire	16
3.1.3	Notion de composante focale	18
3.1.4	Notion d'état focal	19
3.2	Définition d'une relation entre l'ensemble \mathcal{T} des transitions et l'ensemble \mathcal{P} des paramètres	19
3.2.1	Définition informelle de $r^{\mathcal{M}}(\mathcal{T}, \mathcal{P})$	19
3.2.2	Représentation informatique d'une relation formelle entre l'ensemble \mathcal{T} des transitions et l'ensemble \mathcal{P} des paramètres	20
3.2.3	Définition formelle de $transition(\mathcal{M}, t(S, S'), \mathcal{P})$	21

Chapitre 4

Définition de propriétés de réseaux de Thomas

4.1	Propriétés sur la manière de composer des interactions	25
4.1.1	Le langage <i>LComp</i> d'expression de compositions d'interactions	26
4.1.2	Additivité d'une composition d'interactions	29
4.1.3	Observabilité d'une composition d'interactions	30
4.2	Propriétés sur les comportements dynamiques	33
4.2.1	Contraintes d'existence d'un chemin	34
4.2.2	Contraintes d'existence d'un état stationnaire	35
4.2.3	Contraintes d'existence d'un chemin d'atteignabilité	35
4.2.4	Contraintes d'existence d'un cycle	36
4.2.5	Comparaison du pouvoir d'expression de la logique du premier ordre et de la logique CTL dans notre contexte d'étude	37

Fonctionnalités et applications

Chapitre 5

Fonctionnalités de l'outil GNBox

5.1	Relâchement de contraintes et prédictions complètes	41
5.2	Définition de mutants	43

5.3	Introduction dans \mathcal{P} de paramètres d'entrées	44
5.4	Introduction dans \mathcal{P} de paramètres de seuils	45
5.5	Recherche des ensembles cohérents de compositions d'interactions additives et observables	45

Chapitre 6 Applications
--

6.1	Application au contrôle de l'immunité par le phage λ	47
6.1.1	Le modèle et les hypothèses	48
6.1.2	Cohérence et inférence de propriétés	49
6.2	Application au stress nutritionnel carboné chez la bactérie <i>Escherichia coli</i>	51
6.2.1	Le modèle et les hypothèses	51
6.2.2	Relâchement d'hypothèses d'additivité et d'observabilité de composi- tions d'interactions et inférence de propriétés	52
6.3	Application au module <i>gap-gene</i> de la segmentation de l'embryon de <i>Droso-</i> <i>phila melanogaster</i>	54
6.3.1	Le modèle et les hypothèses	55
6.3.2	Nombre minimal de seuils	57

Implémentation

Chapitre 7 Mise en œuvre en PLC de GNBox

7.1	Mise en œuvre de la traduction d'un graphe \mathcal{G} en un modèle \mathcal{M}	63
7.1.1	Représentation informatique d'un graphe \mathcal{G} d'interactions	63
7.1.2	Représentation informatique d'un modèle	64
7.1.3	Le prédicat <i>trans_premodel</i> (G, M)	65
7.2	Mise en œuvre de <i>transition</i> ($\mathcal{M}, ((S, S'), \mathcal{P})$ et <i>path</i> ($\mathcal{M}, Path, \mathcal{P}$)	65
7.3	Mise en œuvre des extensions des fonctionnalités de GNBox	66
7.3.1	Représentation de <i>Mutant</i>	66
7.3.2	Représentation de <i>Modif_nb_thr</i>	66
7.3.3	Représentation de <i>Cplx</i>	67
7.3.4	Représentation de <i>Relax</i>	67
7.4	Mise en œuvre de la production automatique de prédictions complètes	68
7.5	Architecture générale de GNBox	69

Chapitre 8

Intégration d'un solveur SAT à GNBox

8.1	Représentation Booléenne d'une variable entière	72
8.2	Traduction de relations entières en relations Booléennes	72
8.3	Coopération du solveur SAT et du solveur de contraintes PLC dans l'outil GNBox	73
8.4	Performances de l'outil GNBox	75
8.4.1	Recherche de cohérence. Performances	76
8.4.2	Relâchement. Performances	76
8.4.3	Prédictions de propriétés. Performances	76

Conclusion

Chapitre 9

Conclusion

Annexes

Annexe A

Description minimaliste de la Programmation Logique par Contraintes (PLC)

A.1	Aspect déclaratif	84
A.2	Aspect opérationnel	85

Annexe B

**Description continue et description discrète selon R. Thomas de réseaux
génétiques**

Annexe C**Démonstration de la Contrainte 4.3**

C.1 La Contrainte 4.3 est nécessaire	91
C.2 La Contrainte 4.3 est suffisante	92

Annexe D**Fichiers Prolog du graphe d'interaction et du modèle de l'exemple de la Table 3.1**

D.1 Fichier Prolog du graphe d'interaction \mathcal{G} de la Table 3.1	95
D.2 Fichier Prolog du modèle \mathcal{M} de la Table 3.1	95

Annexe E**Equations focales des modèles des deux premières applications du Chapitre 6**

E.1 Equations focales du modèle de contrôle de l'immunité par le phage λ	97
E.2 Equations focales du modèle de stress nutritionnel carboné chez la bactérie <i>Escherichia coli</i>	98

Annexe F**Fichiers Prolog des modules *dynamic_model* et *focal_state***

F.1 Fichier Prolog du module <i>dynamic_model</i>	101
F.2 Fichier Prolog du module <i>focal_state</i>	103

Annexe G**Mise en œuvre du lien entre variables PLC et variables SAT**

Index	109
Bibliographie	111

Introduction

Cette partie débute par un chapitre qui donne le cadre et le sujet de la thèse. Des éléments essentiels précisant le problème sont introduits succinctement afin de clarifier l'approche informatique. Pour terminer nous présentons les travaux principaux qui nous semblent connectés aux nôtres afin de replacer ces derniers dans leur contexte et d'en mieux préciser les objectifs.

Chapitre 1

Introduction

1.1 Cadre

La biologie des systèmes consiste à explorer les effets combinés d'un ensemble d'entités moléculaires et non plus à les explorer entité par entité comme usuellement en biologie moléculaire. Un des objectifs de ce domaine d'étude est la compréhension des phénomènes cellulaires à un niveau moléculaire par la construction de réseaux d'entités se régulant les unes sur les autres. Dans notre cas, nous nous intéressons spécifiquement aux réseaux de régulation génétique exprimés sous la forme de réseaux discrets tels que définis par R. Thomas [51]. Les acteurs sont des (concentrations discrètes de) protéines interagissant sur la production de ces mêmes protéines par l'intermédiaire de gènes codant l'information nécessaire à la production de celles-ci. L'analyse de ces réseaux a pour objectif de déterminer les évolutions possibles de concentrations de ces protéines, et les valeurs possibles de paramètres du réseau satisfaisant des évolutions observées de concentrations. Cette analyse peut se faire "à la main" pour de petits réseaux, par exemple jusqu'à 3 gènes, mais nécessite le recours à des outils mathématiques et informatiques de simulation et d'ingénierie-inverse (*reverse-engineering*) pour maîtriser la complexité de réseaux significatifs d'un point de vue biologique [26].

1.2 Sujet

Notre motivation globale est de fournir au biologiste, étudiant un réseau de régulation génétique, un outil lui permettant de **confronter automatiquement ses hypothèses** pour l'aider à construire son réseau.

L'objectif de cette thèse est de montrer l'intérêt d'une **approche déclarative** pour résoudre ce problème, et de montrer la faisabilité de cette approche à l'aide des technologies Programmation Logique par Contraintes (PLC) [33] et Satisfaction Booléenne (SAT) [38]. Nous entendons par approche déclarative une démarche typique de l'Intelligence Artificielle qui consiste d'une part à formaliser les connaissances (biologiques ici) et d'autre part à construire et analyser, non pas un modèle unique construit "à la main", mais une classe de modèles satisfaisant ces connaissances.

Le principal travail réalisé à l'issue de cette thèse est **la conception et la mise en œuvre d'un outil déclaratif pour l'analyse des réseaux génétiques discrets**. Le programme logique par contraintes résultant de ce travail, nommé GNBox, permet d'utiliser un ensemble partiel d'hypothèses biologiques comme données formelles afin de répondre à de multiples requêtes allant de la simulation au *reverse-engineering*.

Nous précisons dans la Sous-Section 1.2.1 la problématique en présentant certains éléments techniques nécessaires à sa compréhension. La Sous-Section 1.2.2 discute du choix de l’approche informatique utilisée. Finalement, la Sous-Section 1.2.3 illustre à la fois la problématique et l’approche à l’aide d’un exemple simple.

1.2.1 Problématique

Fondamentalement l’outil doit permettre de confronter automatiquement des hypothèses portant à la fois sur les *interactions* du réseau et sur les *comportements*, les comportements étant exprimés par des successions d’états, où un état est un vecteur de niveaux de concentrations des produits des gènes (les protéines) à un temps donné.

Afin de clarifier le problème nous introduisons d’abord les différents éléments sur lequel le problème se pose : un *graphe d’interactions* noté \mathcal{G} offrant des hypothèses partielles sur le réseau de Thomas, un *modèle* noté \mathcal{M} qui décrit précisément le réseau, un *ensemble de paramètres* noté \mathcal{P} , et un *ensemble de transitions* entre états noté \mathcal{T} . Précisons que le mot modèle fait référence dans cette thèse à un ensemble de modèles instanciés, c’est-à-dire dont tous les paramètres sont fixés. Un modèle est donc un “modèle” (au sens commun) mais qui est générique. Ensuite, nous introduisons la notion de relation formelle entre \mathcal{T} et \mathcal{P} vis-à-vis de \mathcal{M} et des règles de transitions.

Grphe \mathcal{G} d’interactions

Un réseau de régulation génétique est représenté de manière abstraite et graphique par un graphe \mathcal{G} d’interactions. Les nœuds de \mathcal{G} sont les gènes et les arêtes orientées sont les interactions entre ces gènes. Un entier représentant un seuil de concentration d’activation de l’interaction étiquète chaque arête¹.

Une arête d’un gène a vers un gène b avec un seuil t autorise une différence d’évolution de la concentration de la protéine b entre des états adjacents de part et d’autre du seuil t . En d’autre termes, cette arête indique que l’évolution de la concentration de la protéine b peut être influencée par la position respective du seuil t et de la concentration de la protéine a . D’une part, une telle arête n’impose pas une différence d’évolution. D’autre part l’absence d’une telle arête indique l’absence d’une différence d’évolution.

Modèle \mathcal{M}

Un réseau de régulation génétique discret est spécifié par un système d’équations discrètes définissant la direction d’évolution de chaque concentration dans un état donné. Ce système d’équations provient usuellement de la discrétisation d’un système d’équations différentielles linéaires par morceaux couramment associé à un réseau de régulation (voir Annexe B²). Les paramètres du système d’équations discrètes n’étant pas systématiquement connus, nous associons au système d’équations un ensemble de contraintes sur ses paramètres. Le couple ainsi formé d’un système d’équations et d’un ensemble de contraintes est appelé modèle \mathcal{M} . Un tel modèle définit une classe de modèles instanciés (c’est-à-dire dont tous les paramètres sont fixés).

Dans cette thèse nous déduisons un modèle \mathcal{M} à partir d’un graphe \mathcal{G} d’interactions (voir Section 3.1).

¹Usuellement les arêtes des graphes d’interactions sont aussi étiquetées par un signe [51]. Nous choisissons ce type primitif de graphes d’interactions afin de simplifier et généraliser (vis-à-vis des signes) la formalisation (voir Section 3.1) et faciliter l’expression d’hypothèses sur la manière de composer des interactions (voir Section 4.1).

²L’Annexe B ne peut être comprise dans son intégralité qu’après lecture du Chapitre 3 et de la Section 4.1.

Ensemble \mathcal{P} de paramètres et ensemble \mathcal{T} de transitions

A un modèle \mathcal{M} sont associés deux ensembles interdépendants et de nature différente :

- un ensemble \mathcal{P} de paramètres. Il s’agit d’une version discrète des paramètres cinétiques réels (apparaissant dans les équations différentielles) gouvernant les comportements du réseau³. D’autres paramètres sont aussi considérés dans un second temps, dont les seuils (voir Section 5.4).
- un ensemble \mathcal{T} de transitions qui exprime les successions possibles d’états (les comportements du réseau).

L’interdépendance de ces deux ensembles provient du fait qu’une propriété sur l’un de ces deux ensembles (\mathcal{P} ou \mathcal{T}) induit une propriété sur l’autre ensemble⁴.

Clarification de la problématique

Nous souhaitons fournir une aide à la construction de réseaux génétiques en proposant un outil efficace, simple d’utilisation et facilement extensible pour confronter des hypothèses biologiques. Ces hypothèses portent soit directement soit indirectement sur \mathcal{T} et \mathcal{P} .

Les problèmes qui se posent sont les suivants :

1. définir une relation formelle entre \mathcal{T} et \mathcal{P} , notée $r^{\mathcal{M}}(\mathcal{T}, \mathcal{P})$, vraie si \mathcal{T} est l’ensemble de transitions du modèle \mathcal{M} ayant comme ensemble de paramètres \mathcal{P} ⁵. Le fait d’introduire une relation entre \mathcal{T} et \mathcal{P} et non une fonction (par exemple donnant \mathcal{T} pour un \mathcal{P} fixé) est révélateur de l’approche déclarative qui permet dans notre cas d’introduire et de déduire de l’information à la fois sur \mathcal{T} et sur \mathcal{P} .
2. définir des langages d’expression d’hypothèses d’un niveau d’abstraction plus élevé, c’est-à-dire ne portant pas directement sur \mathcal{T} et \mathcal{P} , afin d’éviter à l’utilisateur un travail fastidieux de transformation de ces hypothèses en terme de \mathcal{T} et \mathcal{P} et vice-versa.
3. prévoir une large paramétrisation de la construction et de la représentation d’un modèle pour exprimer un maximum de type de requêtes.
4. construire un outil déclaratif à partir de ces notions.
5. montrer la faisabilité et valider l’approche déclarative sur des problèmes biologiques réels.

1.2.2 Approche informatique

Nous proposons pour résoudre les problèmes cités ci-dessus la technologie *Programmation Logique par Contraintes* (PLC) avec l’utilisation d’un *solveur sur domaines finis* et l’intégration d’un *solveur SAT*.

La PLC est un paradigme de programmation fondé sur la logique du premier ordre et utilisant les clauses de Horn. De plus, la PLC considère de façon spécifique certaines classes de termes logiques et propose des résolutions efficaces d’équations sur ces termes (contraintes). Nous choisissons ici d’utiliser un solveur de contraintes sur domaines finis, c’est-à-dire sur des termes qui

³Plus précisément il s’agit de rapports de paramètres cinétiques.

⁴De manière plus spécifique si nous fixons \mathcal{P} alors \mathcal{T} est automatiquement fixé. Par contre nous verrons qu’un même \mathcal{T} instancié peut correspondre à plusieurs valuations de \mathcal{P} (voir Exemple 4.10).

⁵Nous verrons dans la Section 3.2 que pour un souci de taille de représentation de cette relation nous implémentons en fait une relation formelle entre une transition Tr de \mathcal{T} et \mathcal{P} . Cette relation est vraie si Tr est une transition du modèle \mathcal{M} ayant comme ensemble de paramètres \mathcal{P} .

sont des entiers appartenant à des domaines finis. Les contraintes posées sont donc des contraintes sur variables entières.

Un programme en PLC est une formule logique, et nous pouvons considérer que son “exécution” est la construction des preuves de cette formule. Un programme en PLC est alors “réversible” dans le sens où il permet d’imposer et d’obtenir de l’information partielle sur toutes les variables constituant la formule du programme. Par ailleurs, la puissance d’expression de la logique du premier ordre et celle des contraintes sur entiers sont susceptibles *a priori* de permettre la définition de propriétés et de fonctionnalités très générales. Enfin, comme toutes les variables de \mathcal{P} et \mathcal{T} ont des domaines entiers et étant donné l’aspect discret de l’analyse à mettre en place, l’utilisation d’un solveur de contraintes sur entiers apparaît *a priori* tout-à-fait adapté.

La PLC semble pour ces raisons adéquate pour résoudre nos problèmes. C’est ce point qui est défendu et étayé par cette thèse. Pour les lecteurs non familier avec la PLC nous en donnons une brève description en Annexe A.

Pour des raisons d’efficacité de résolution, nous proposons aussi d’intégrer un solveur SAT à l’outil basé sur la PLC. Un solveur SAT permet de tester la consistance d’une formule Booléenne sous forme normale conjonctive (CNF)⁶. Le choix d’utiliser un solveur SAT est dû d’une part aux performances surprenantes de ces solveurs sur un grand nombre de problèmes (dont la vérification de programmes, la conception de circuits et l’ordonnancement) et d’autre part à la taille raisonnable, dans notre cas, de la traduction des contraintes PLC en formules CNF (voir Section 8.2). Un excellent rapport comparatif des technologies SAT et programmation par contraintes en terme de résolution et de modélisation se trouve en [9].

1.2.3 Illustration du problème et de l’approche à l’aide d’un exemple

Nous présentons dans cette section un exemple simplifié permettant d’illustrer aisément la problématique et l’approche PLC utilisée.

Exemple 1.1 *Soit un modèle \mathcal{M} doté de l’ensemble \mathcal{P} de paramètres et de l’ensemble \mathcal{T} de transitions. L’ensemble \mathcal{T} de transitions fait intervenir uniquement deux états s_A et s_B . L’ensemble \mathcal{P} de paramètres contient un unique paramètre, noté K , qui prend une valeur entière dans l’intervalle $[0, 2]$. Soit la relation $r^{\mathcal{M}}(\mathcal{T}, \mathcal{P})$ vraie si \mathcal{T} est conforme à \mathcal{M} doté de l’ensemble de paramètres \mathcal{P} . Supposons que $r^{\mathcal{M}}(\mathcal{T}, \mathcal{P})$ soit donnée par la formule logique suivante :*

$$r^{\mathcal{M}}(\mathcal{T}, \mathcal{P}) \Leftrightarrow (\mathcal{P} = \{K\}) \wedge ((K = 0 \wedge \mathcal{T} = \{s_A \rightarrow s_A, s_B \rightarrow s_A\}) \vee \\ (K = 1 \wedge \mathcal{T} = \{s_A \rightarrow s_A, s_B \rightarrow s_A, s_B \rightarrow s_B\}) \vee \\ (K = 2 \wedge \mathcal{T} = \{s_A \rightarrow s_B, s_B \rightarrow s_A\}))$$

où $s \rightarrow s'$ représente une transition de l’état s vers l’état s' . La Figure 1.1 représente la formule $r^{\mathcal{M}}(\mathcal{T}, \mathcal{P})$ comme la disjonction de 3 cas de valuation de \mathcal{P} et \mathcal{T} où \mathcal{T} est représenté comme un graphe de transitions entre les états s_A et s_B .

Résoudre une requête \mathcal{Q} pour le modèle \mathcal{M} consiste à vérifier la consistance de la formule logique $r^{\mathcal{M}}(\mathcal{T}, \mathcal{P}) \Rightarrow \mathcal{Q}$. L’ensemble des solutions est exprimé sous forme d’une expression logique qui indique les valeurs possibles des variables mises en jeu dans la requête et le modèle. Les trois requêtes \mathcal{Q}_i suivantes et résultats associés \mathcal{R}_i sont exprimés à la fois en langage naturel et de

⁶Ce format d’entrée des solveurs SAT est explicité en début de Chapitre 8.

$$\begin{array}{c}
\mathcal{P} = \{0\} \\
\mathcal{T} = \begin{array}{c} \curvearrowright \\ s_A \leftarrow s_B \end{array}
\end{array}
\quad \left| \quad
\begin{array}{c}
\mathcal{P} = \{1\} \\
\mathcal{T} = \begin{array}{c} \curvearrowright \\ s_A \leftarrow s_B \curvearrowright \end{array}
\end{array}
\quad \left| \quad
\begin{array}{c}
\mathcal{P} = \{2\} \\
\mathcal{T} = s_A \overset{\curvearrowright}{\rightleftarrows} s_B
\end{array}$$

FIG. 1.1 – Relation $r^{\mathcal{M}}(\mathcal{T}, \mathcal{P})$ exprimée comme une disjonction de 3 valuations de \mathcal{T} et \mathcal{P} .

manière formelle. Elles illustrent successivement une requête de recherche des états stationnaires, une requête d'inférence de valeurs de paramètres exploitant l'existence d'un comportement fixé, et une requête d'inférence exploitant une connaissance comportementale partielle.

1. – \mathcal{Q}_1 : Le modèle \mathcal{M} comporte-t-il des états stationnaires et lesquels ?
 - \mathcal{Q}_1 : $\exists S, K \quad S \rightarrow S \in \mathcal{T}$
 - \mathcal{R}_1 : $(S = s_A \wedge K \in [0, 1]) \vee (S = s_B \wedge K = 1)$
 - \mathcal{R}_1 : Les états stationnaires possibles sont s_A et s_B . Nous obtenons de plus les informations : s_A est stationnaire si et seulement si $K \in [0, 1]$ et s_B est stationnaire si et seulement si $K = 1$.
2. – \mathcal{Q}_2 : Quelles sont les valeurs des paramètres de \mathcal{P} qui admettent la séquence d'états $s_A \rightarrow s_B \rightarrow s_A$?
 - \mathcal{Q}_2 : $\exists K \quad s_A \rightarrow s_B \in \mathcal{T} \wedge s_B \rightarrow s_A \in \mathcal{T}$
 - \mathcal{R}_2 : $K = 2$
 - \mathcal{R}_2 : Seule la valuation \mathcal{P} telle que $K = 2$ admet le comportement $s_A \rightarrow s_B \rightarrow s_A$.
3. – \mathcal{Q}_3 : Quelles sont les séquences admises par le modèle \mathcal{M} qui sont de la forme $S' \rightarrow S \rightarrow S$ et ayant la propriété $S' \neq S$?
 - \mathcal{Q}_3 : $\exists S', S, K \quad S' \rightarrow S \in \mathcal{T} \wedge S \rightarrow S \in \mathcal{T} \wedge S' \neq S$
 - \mathcal{R}_3 : $S' = s_B \wedge S = s_A \wedge K \in [0, 1]$
 - \mathcal{R}_3 : Seule la séquence d'états $s_B \rightarrow s_A \rightarrow s_A$ répond aux contraintes de la question, avec l'information additionnelle $K \in [0, 1]$.

L'exemple 1.1 donne une idée, sans même citer des propriétés biologiques, à la fois de la palette de questions que nous souhaitons résoudre et de la manière commune de les résoudre en passant par une représentation des réponses comme les solutions d'un ensemble de contraintes.

1.3 Plan

Le Chapitre 2 discute des travaux connectés aux nôtres afin de replacer ces derniers dans leur contexte et d'en préciser les objectifs. Afin de montrer que la PLC est adéquate pour résoudre les problèmes listés en Sous-Section 1.2.1, nous procédons de la manière suivante :

1. dans le Chapitre 3, nous définissons un modèle \mathcal{M} et la relation $r^{\mathcal{M}}(\mathcal{T}, \mathcal{P})$ sous forme de contraintes .
2. dans le Chapitre 4, nous définissons des langages de haut niveau facilitant l'expression de propriétés usuelles sur la manière de composer les interactions et sur les comportements. Leur sémantique est décrite à l'aide de contraintes.
3. dans le Chapitre 5, nous prévoyons une large paramétrisation de la représentation d'un modèle \mathcal{M} pour exprimer un large choix de types de requêtes. En particulier nous montrons comment assurer la cohérence de l'ensemble des hypothèses, l'identification de comportements spécifiques et la prédiction de propriétés comme les contraintes à relâcher dans le cas

d'une inconsistance. Les hypothèses peuvent alors porter de plus sur plusieurs modèles similaires qui diffèrent par des mutations de certains gènes et sur des connaissances partielles sur les ordres des seuils de discrétisation.

4. dans le Chapitre 7 nous présentons la mise en œuvre de l'outil GNBox implémentant les notions des précédents points. Le chapitre 8 est consacré à la conception et la réalisation de l'intégration d'un solveur SAT dans GNBox.
5. dans le Chapitre 6, nous validons l'approche déclarative à l'aide d'applications biologiques réelles⁷ portant sur le contrôle de l'immunité chez le bactériophage λ , le stress nutritionnel carboné chez *E. coli*, et le contrôle par les gènes *gap* de la segmentation de l'embryon de la mouche *Drosophila melanogaster*. La faisabilité de cette approche est illustrée par les performances de GNBox à des requêtes d'une importante complexité combinatoire.

Finalement, le Chapitre 9 dresse un bilan de cette thèse et propose de futures développements.

⁷Par souci d'introduire le plus tôt possible un maximum d'exemples illustrant les notions introduites dans les Chapitres 3, 4 et 5 nous avons délibérément placé à la suite de ces chapitres le Chapitre 6 sur les applications biologiques.

Chapitre 2

Travaux connexes

Nous présentons dans cette section les principaux travaux dont l'objectif est l'aide à l'analyse des réseaux génétiques discrets tels que définis par R. Thomas. Nous choisissons de les regrouper en deux grandes catégories selon qu'ils considèrent que le modèle est connu (*i.e.* avec des paramètres fixés), ou non.

2.1 Analyse de réseaux dont les paramètres sont fixés

Si les paramètres sont connus deux objectifs sont poursuivis : identifier des propriétés comportementales en procédant par simulation, et vérifier qu'une propriété comportementale est acceptée par le modèle.

La problématique simulation de comportements pour les réseaux de Thomas est abordée essentiellement dans les travaux suivants.

H. de Jong *et al.* [24, 25, 5] ont conçu le logiciel GNA de modélisation et simulation qui utilise des équations différentielles linéaires par morceaux à partir desquelles sont produites les mêmes équations discrètes que celles des réseaux de Thomas. Cependant ce logiciel prend en compte les états positionnés sur les seuils de discrétisation, dits *états singuliers*. Cette extension est nécessaire pour certains réseaux où l'atteignabilité d'états ne peut se faire qu'en "glissant" le long des seuils de discrétisation.

D. Thieffry *et al.* [49] ont conçu le logiciel GINSim [31] de modélisation et simulation de réseaux de Thomas. Les fonctionnalités de simulation offertes par ce logiciel exploitent des propriétés sur les circuits du graphe d'interactions.

Des approches de type simulation, mais basées sur le formalisme des réseaux de Petri (conçu à l'origine pour analyser les systèmes concurrents complexes), sont aussi explorées dans plusieurs recherches dont celles de C. Chaouiya [13], L. J. Steggle [47], et S. Troncale [54]. Les réseaux de Petri présentent un aspect non déterministe et peuvent traiter des données incomplètes.

L'identification des états stationnaires est une problématique essentielle traitée de maintes manières et en particulier en utilisant des abstractions proches du formalisme de R. Thomas pour ensuite repasser à des formalismes continus en tenant compte des résultats obtenus dans un formalisme discret. Ces méthodes considèrent les propriétés nécessaires à la multistationarité et à la stabilité sur les circuits du graphe d'interactions [51, 53, 52].

V. Devloo [27] utilise, comme nous, la modélisation sous forme d'un CSP (*Constraint Satis-*

faction Problem)⁸. Son objectif est d'identifier les états stationnaires, potentiellement singuliers, de grands réseaux de Thomas dont les paramètres sont tous connus. Les performances obtenues sont impressionnantes.

A. Naldi *et al.* utilisent des diagrammes de décisions multi-valués pour représenter les fonctions de transitions pour chaque noeud afin, entre autres, d'identifier les états stationnaires [37]. Cette structure de données est utilisé dans le logiciel GINSim.

L. Mendoza *et al.* travaillent sur le logiciel SQUAD de simulation de réseaux de signalisation utilisant une approche qualitative afin de trouver les états stationnaires stables pour ensuite résoudre numériquement le système continu [36, 11]; cette méthode pouvant s'appliquer à des réseaux en manque d'information stoichiométriques.

Le second objectif, c'est-à-dire la vérification qu'une propriété comportementale est acceptée nécessite d'abord le choix d'un langage adapté pour l'exprimer. La majeure partie de travaux choisissent la logique temporelle CTL (*Computational Tree Logic*) [16, 39] qui a été introduite initialement pour exprimer des propriétés de comportement de programme informatique.

C'est le cas de G. Bernot *et al.* [8, 41, 40] et de N. Chabrier et F. Fages [12]. Les logiciels GNA et GINSim intègrent aussi CTL. La vérification de propriétés exprimées en CTL est assurée à l'aide d'algorithmes dits de *model-checking* (par exemple NuSMV [14]) qui ont fait l'objet de très nombreuses études.

2.2 Analyse de réseaux dont les paramètres ne sont pas fixés

Si les paramètres ne sont pas connus il faut travailler avec des classes de modèles qui satisfont à un ensemble de propriétés. Deux objectifs sont alors poursuivis : vérifier si une propriété est satisfaite par au moins un modèle de la classe, prédire une propriété satisfaite pour tous les modèles de la classe. Notons que dans ce dernier cas il est nécessaire de définir un sous-ensemble de propriétés intéressantes pour des raisons de performances si la propriété à prédire n'est pas fixée.

Les travaux les plus significatifs dans cette optique sont ceux de G. Bernot *et al.* Ils consistent à déterminer en "extension" c'est-à-dire sous la forme d'un ensemble de valeurs, la classe des modèles satisfaisant des propriétés comportementales. Ces propriétés sont exprimées en CTL. Algorithmiquement, le processus consiste à produire, avec des optimisations, tous les modèles instanciés possibles et à retenir, à l'aide d'un algorithme de *model-checking*, tous ceux qui satisfont les propriétés CTL.

Le travail du regretté D. Mateus [35] tend plus vers une représentation en "intention" de la classe de modèle c'est-à-dire comme solutions d'un ensemble de contraintes. Celle-ci est obtenue à l'issue de deux étapes. La première consiste, pour chaque comportement répondant à une formule LTL (*Linear Temporal Logic*, une logique temporelle proche de CTL) à déterminer la conjonction d'inégalités de paramètres qui est impliquée par ce comportement. Elle utilise un *model-checker* dit symbolique. La seconde étape a pour but de simplifier la disjonction de ces conjonctions. Elle utilise un outil de simplifications de formules Booléennes. Au total nous notons un coût important du aux énumérations soit des modèles soit des comporements. Un de nos objectifs est d'y remédier.

F. Fages *et al.* travaillent sur l'environnement de modélisation BIOCHAM pour analyser et vérifier la structure de modèles en cours de construction, tout en fournissant avec une approche

⁸La programmation par contraintes concerne la formulation d'une CSP et sa résolution [4]

contraintes des valeurs de paramètres cinétiques discrets appropriées pour reproduire des comportements observés, et cela pour de multiples types d'information qualitative ou quantitative [12, 10].

2.3 Comparaison avec notre proposition

Bien sûr, nous nous attaquons à l'analyse de réseaux dont les paramètres ne sont pas fixés *a priori*. Notre approche se distingue d'abord par une formulation en "intension" de la classe des modèles satisfaisant aux hypothèses. En effet une formulation en "extension" pose deux sortes de problèmes. D'une part, elle est très peu lisible par un utilisateur. D'autre part, elle est très peu propre à un bon traitement algorithmique de la prédiction de propriétés. Nous sommes amené à un problème d'apprentissage "à partir d'exemples" dont la résolution est coûteuse.

Ensuite, notre approche se veut plus ambitieuse.

- d'un point de vue satisfaisabilité. Précisons que vérifier la satisfaisabilité d'une contrainte par rapport à un ensemble de contraintes revient à ajouter cette contrainte à l'ensemble et à vérifier la cohérence du nouvel ensemble. Non seulement notre objectif est d'obtenir de bonnes performances pour cette vérification mais de plus nous proposons des recours au cas où cette vérification mène à une incohérence.
- d'un point de vue prédiction. Précisons que prédire une propriété à partir d'un ensemble de contraintes revient à ajouter la négation de la propriété à cet ensemble et à vérifier l'inconsistance du nouvel ensemble. Il s'agit d'obtenir de bonnes performances pour l'obtention de propriétés appartenant à des langages se situant à un niveau plus haut que celui des paramètres.
- d'un point de vue expression de propriétés. Certes exprimer les comportements à partir de formules CTL est intéressant. Cependant, notre objectif est de proposer un plus grand pouvoir d'expression et comme il a été dit un niveau d'expression plus élevé.

Nos travaux qui suivent cette optique sont (dans l'ordre chronologique) [29, 21, 18, 22, 20, 23, 19].

Formalisation du problème

Cette partie débute avec le Chapitre 3 définissant le formalisme de Thomas à l'aide de contraintes. Plus précisément, il définit un modèle \mathcal{M} à partir d'un graphe \mathcal{G} d'interactions et la relation $r^{\mathcal{M}}(\mathcal{T}, \mathcal{P})$ liant formellement l'ensemble \mathcal{T} de transitions et l'ensemble \mathcal{P} de paramètres d'un modèle \mathcal{M} .

La suite de cette partie propose des langages de plus haut niveau que celui des contraintes entières sur les variables de \mathcal{T} et \mathcal{P} , pour faciliter l'expression de propriétés usuelles sur la manière de composer les interactions et sur les comportements. La sémantique de ces langages est donnée en terme de contraintes sur les paramètres de \mathcal{P} .

Chapitre 3

Formalisation des réseaux de Thomas en contraintes

La formalisation des réseaux de Thomas nécessite deux définitions : celle d'un modèle \mathcal{M} à partir d'un réseau génétique représenté de manière abstraite et graphique par un graphe \mathcal{G} d'interactions (présentée en Section 3.1), et celle de la relation entre l'ensemble \mathcal{T} de transitions et l'ensemble \mathcal{P} de paramètres (présentée en Section 3.2). Cette partie de la thèse montre que les réseaux de Thomas peuvent aisément être à la fois précisément formalisés et implémentés à l'aide de contraintes.

3.1 Définition d'un modèle \mathcal{M}

Nous présentons dans cette section la définition précise d'un modèle \mathcal{M} associé à un graphe \mathcal{G} d'interactions. Quatre notions fondamentales successives sont nécessaires pour définir \mathcal{M} . Elles sont présentées en Sous-Sections 3.1.1, 3.1.2, 3.1.3 et 3.1.4, et elles sont illustrées par l'exemple donné par la Table 3.1 (la première colonne faisant référence à la Sous-Section associée à la notion illustrée).

Donnons pour commencer les notations élémentaires. Soit S_c la concentration du produit (protéine ou ARN régulateur) du gène c dans l'état S exprimé comme un entier compris entre 0 et max_c , max_c étant la valeur maximale de concentration du produit du gène c .

Soit t_c^p le seuil de c d'indice p , où les indices p prennent les valeurs de 1 à max_c . Nous choisissons de considérer, dans un premier stade, les variables t_c^p égales à p^9 .

Soit un système à n composantes c_i , $i \in [1, n]$, dont les concentrations prennent des valeurs de 0 à max_{c_i} . L'espace d'états contient alors $\prod_{c_i} (max_{c_i} + 1)$ états. Chaque état S est représenté par un vecteur de concentrations discrètes de toutes les composantes dans cet état $S : \langle S_{c_1}, \dots, S_{c_n} \rangle$.

Exemple 3.1 *Considérons l'exemple de graphe \mathcal{G} d'interactions de la troisième ligne de la Table 3.1. Ce graphe contient $n = 2$ composantes nommées x et y avec $max_x = 1$ et $max_y = 2$, car l'espace de concentration pour la composante x est découpé en deux régions par le seuil $t_x^1 = 1$ et l'espace de concentration pour la composante y est découpé en trois régions par les seuils $t_y^1 = 1$*

⁹Nous verrons dans la Section 5.4 qu'elles sont considérées effectivement comme des variables (non instanciées systématiquement). Nous présentons dans la Section 6.3 quelques requêtes avec des variables non instanciées pour les valeurs de seuils.

et $t_y^2 = 2$. Finalement, l'espace des concentrations est découpé en six régions. Ces régions sont les états $S = \langle S_x, S_y \rangle$ possibles du système : $\langle 0, 0 \rangle$, $\langle 0, 1 \rangle$, $\langle 0, 2 \rangle$, $\langle 1, 0 \rangle$, $\langle 1, 2 \rangle$ et $\langle 1, 2 \rangle$.

3.1.1 Notion d'interaction et d'activité d'une interaction

Définition 3.1 Une *interaction* de \mathcal{G} est un triplet $(c', t_{c'}^p, c)$ associé à une arête du nœud c' vers le nœud c dans \mathcal{G} et décorée par un entier $t_{c'}^p$ dénotant le seuil à partir duquel l'interaction peut se produire.

Chaque interaction $(c', t_{c'}^p, c)$ sur la cible c est associée à un indice r unique, $1 \leq r \leq r_c$, où r_c est le nombre d'arêtes entrantes dans le nœud c dans le graphe \mathcal{G} .

Nous notons i_c^r l'interaction sur c d'indice r .

Définition 3.2 L'*activité d'une interaction* $i_c^r = (c', t_{c'}^p, c)$ dans l'état S est $S_{c'} \geq t_{c'}^p$. Elle est notée $I_{c,S}^r$.

Exemple 3.2 Pour le nœud y dans le graphe \mathcal{G} de la Table 3.1 nous obtenons $I_{y,S}^1 \equiv S_x \geq t_x^1$ pour l'interaction $i_y^1 = (x, t_x^1, y)$ que nous avons choisi d'indice 1, et $I_{y,S}^2 \equiv S_y \geq t_y^2$ pour l'interaction $i_y^2 = (y, t_y^2, y)$ que nous avons choisi d'indice 2. De manière similaire, pour le nœud x , l'interaction $i_x^1 = (y, t_y^1, x)$ est active dans l'état S si $I_{x,S}^1$, défini par $\equiv S_y \geq t_y^1$, est vrai.

Il est à noter que nous autorisons plusieurs arêtes de c' vers c décorées par des seuils différents indiquant, d'un point de vue biologique, que le gène c' peut exercer plusieurs effets sur c à des niveaux de concentration différents. De plus un même seuil $t_{c'}^p$ peut apparaître sur plus d'une flèche du graphe \mathcal{G} entre couples de nœuds qui diffèrent par leur nœud cible.

3.1.2 Notion de contexte cellulaire

L'ensemble des interactions sur une cible c , soit $\{i_c^r | 1 \leq r \leq r_c\}$, induit une partition de l'espace des concentrations selon les seuils des interactions. Chaque sous-ensemble d'interactions identifie une région (potentiellement vide) de l'espace d'états appelée *contexte cellulaire* que nous définissons précisément par la Définition 3.3. Avant cela nous caractérisons un sous-ensemble d'interactions par un vecteur Booléen V tel que la $r^{\text{ème}}$ composante de V est vraie si et seulement si l'interaction i_c^r est présente dans le sous-ensemble d'interactions considéré.

Définition 3.3 Soit V un vecteur de Booléens de r_c éléments. Soit V_r l'élément d'indice r de V . Le *contexte cellulaire* associé au sous-ensemble d'interactions caractérisé par V est la région composée des états S tels que

$$\bigwedge_{r=1}^{r_c} (V_r \Leftrightarrow I_{c,S}^r)$$

Nous identifions un contexte cellulaire par cette dernière condition, notée $\text{Cell}_{c,S}^l$, où $l \in [1, l_c]$ avec $l_c = 2^{r_c}$ le nombre de sous-ensembles d'interactions pour c , et $l - 1$ égal à la valeur décimale du nombre binaire constitué des composantes de V (par exemple $V = (1, 0)$ correspond à $l = 3$, $l - 1$ étant égal à 2).

Exemple 3.3 Dans le graphe \mathcal{G} de la Table 3.1 les quatre sous-ensembles d'interactions ayant pour cibles y sont $\{\}$, $\{i_y^2\}$, $\{i_y^1\}$ et $\{i_y^1, i_y^2\}$ correspondant aux quatre vecteurs V suivant : $(0, 0)$, $(0, 1)$, $(1, 0)$ et $(1, 1)$. Comme il n'y a qu'une arête entrante sur le nœud x nous avons seulement

3.1. Définition d'un modèle \mathcal{M}

Section		relatif à x et y																																																																																	
		relatif à x	relatif à y																																																																																
	\mathcal{G}																																																																																		
3.1.1	i_c^r $I_{c,S}^r$	$i_x^1 = (y, t_y^1, x) : I_{x,S}^1 = S_y \geq t_y^1$	$i_y^1 = (x, t_x^1, y) : I_{y,S}^1 = S_x \geq t_x^1$ $i_y^2 = (y, t_y^2, y) : I_{y,S}^2 = S_y \geq t_y^2$																																																																																
3.1.2	$Cell_{c,S}^l$	$Cell_{x,S}^1 = -I_{x,S}^1 = S_y < t_y^1$ $Cell_{x,S}^2 = I_{x,S}^1 = S_y \geq t_y^1$	$Cell_{y,S}^1 = -I_{y,S}^1 \wedge -I_{y,S}^2 = S_x < t_x^1 \wedge S_y < t_y^2$ $Cell_{y,S}^2 = -I_{y,S}^1 \wedge I_{y,S}^2 = S_x < t_x^1 \wedge S_y \geq t_y^2$ $Cell_{y,S}^3 = I_{y,S}^1 \wedge -I_{y,S}^2 = S_x \geq t_x^1 \wedge S_y < t_y^2$ $Cell_{y,S}^4 = I_{y,S}^1 \wedge I_{y,S}^2 = S_x \geq t_x^1 \wedge S_y \geq t_y^2$																																																																																
3.1.3	K_c^l \mathcal{P}	K_x^1, K_x^2	$K_y^1, K_y^2, K_y^3, K_y^4$																																																																																
	$F_{c,S}$	$F_{x,S} = \sum_m K_x^m \cdot Cell_{x,S}^m$ $= K_x^1 \cdot S_y < t_y^1 +$ $K_x^2 \cdot S_y \geq t_y^1$	$F_{y,S} = \sum_m K_y^m \cdot Cell_{y,S}^m$ $= K_y^1 \cdot S_x < t_x^1 \wedge S_y < t_y^2 +$ $K_y^2 \cdot S_x < t_x^1 \wedge S_y \geq t_y^2 +$ $K_y^3 \cdot S_x \geq t_x^1 \wedge S_y < t_y^2 +$ $K_y^4 \cdot S_x \geq t_x^1 \wedge S_y \geq t_y^2$																																																																																
3.1.4	F_S \mathcal{M}	$F_S = \langle F_{x,S}, F_{y,S} \rangle$																																																																																	
	$r^{\mathcal{M}}(\mathcal{T}, \mathcal{P})$	<table style="width:100%; border-collapse: collapse;"> <tr> <td style="width:15%;">$(\langle 0,0 \rangle \rightarrow \langle 0,0 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td style="width:35%;">$K_x^1 = 0 \wedge K_y^1 = 0$</td> <td style="width:15%;">$) \wedge$</td> <td style="width:35%;"></td> </tr> <tr> <td>$(\langle 0,0 \rangle \rightarrow \langle 0,1 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^1 > 0$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 0,0 \rangle \rightarrow \langle 1,0 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^1 > 0$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 0,1 \rangle \rightarrow \langle 0,0 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^2 = 0 \wedge K_y^1 < 1$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 0,1 \rangle \rightarrow \langle 0,1 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^2 = 0 \wedge K_y^1 = 1$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 0,1 \rangle \rightarrow \langle 0,2 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^2 > 0 \wedge K_y^1 > 1$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 0,1 \rangle \rightarrow \langle 1,1 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^2 > 0$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 0,2 \rangle \rightarrow \langle 0,1 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^2 > 0 \wedge K_y^2 < 2$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 0,2 \rangle \rightarrow \langle 0,2 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^2 > 0 \wedge K_y^2 = 2$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 0,2 \rangle \rightarrow \langle 1,2 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^2 > 0$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 1,0 \rangle \rightarrow \langle 0,0 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^1 < 1$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 1,0 \rangle \rightarrow \langle 1,0 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^1 = 1 \wedge K_y^3 = 0$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 1,0 \rangle \rightarrow \langle 1,1 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^1 = 1 \wedge K_y^3 > 0$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 1,1 \rangle \rightarrow \langle 0,1 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^2 < 1$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 1,1 \rangle \rightarrow \langle 1,0 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^2 < 1$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 1,1 \rangle \rightarrow \langle 1,1 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^2 = 1 \wedge K_y^3 = 1$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 1,1 \rangle \rightarrow \langle 1,2 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^2 = 1 \wedge K_y^3 > 1$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 1,2 \rangle \rightarrow \langle 0,2 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^2 < 1$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 1,2 \rangle \rightarrow \langle 1,1 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^2 < 1$</td> <td>$) \wedge$</td> <td></td> </tr> <tr> <td>$(\langle 1,2 \rangle \rightarrow \langle 1,2 \rangle \in \mathcal{T} \Leftrightarrow$</td> <td>$K_x^2 = 1 \wedge K_y^4 = 2$</td> <td>$) \wedge$</td> <td></td> </tr> </table>		$(\langle 0,0 \rangle \rightarrow \langle 0,0 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^1 = 0 \wedge K_y^1 = 0$	$) \wedge$		$(\langle 0,0 \rangle \rightarrow \langle 0,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^1 > 0$	$) \wedge$		$(\langle 0,0 \rangle \rightarrow \langle 1,0 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^1 > 0$	$) \wedge$		$(\langle 0,1 \rangle \rightarrow \langle 0,0 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 = 0 \wedge K_y^1 < 1$	$) \wedge$		$(\langle 0,1 \rangle \rightarrow \langle 0,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 = 0 \wedge K_y^1 = 1$	$) \wedge$		$(\langle 0,1 \rangle \rightarrow \langle 0,2 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 > 0 \wedge K_y^1 > 1$	$) \wedge$		$(\langle 0,1 \rangle \rightarrow \langle 1,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 > 0$	$) \wedge$		$(\langle 0,2 \rangle \rightarrow \langle 0,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 > 0 \wedge K_y^2 < 2$	$) \wedge$		$(\langle 0,2 \rangle \rightarrow \langle 0,2 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 > 0 \wedge K_y^2 = 2$	$) \wedge$		$(\langle 0,2 \rangle \rightarrow \langle 1,2 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 > 0$	$) \wedge$		$(\langle 1,0 \rangle \rightarrow \langle 0,0 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^1 < 1$	$) \wedge$		$(\langle 1,0 \rangle \rightarrow \langle 1,0 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^1 = 1 \wedge K_y^3 = 0$	$) \wedge$		$(\langle 1,0 \rangle \rightarrow \langle 1,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^1 = 1 \wedge K_y^3 > 0$	$) \wedge$		$(\langle 1,1 \rangle \rightarrow \langle 0,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 < 1$	$) \wedge$		$(\langle 1,1 \rangle \rightarrow \langle 1,0 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 < 1$	$) \wedge$		$(\langle 1,1 \rangle \rightarrow \langle 1,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 = 1 \wedge K_y^3 = 1$	$) \wedge$		$(\langle 1,1 \rangle \rightarrow \langle 1,2 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 = 1 \wedge K_y^3 > 1$	$) \wedge$		$(\langle 1,2 \rangle \rightarrow \langle 0,2 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 < 1$	$) \wedge$		$(\langle 1,2 \rangle \rightarrow \langle 1,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 < 1$	$) \wedge$		$(\langle 1,2 \rangle \rightarrow \langle 1,2 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 = 1 \wedge K_y^4 = 2$	$) \wedge$	
$(\langle 0,0 \rangle \rightarrow \langle 0,0 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^1 = 0 \wedge K_y^1 = 0$	$) \wedge$																																																																																	
$(\langle 0,0 \rangle \rightarrow \langle 0,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^1 > 0$	$) \wedge$																																																																																	
$(\langle 0,0 \rangle \rightarrow \langle 1,0 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^1 > 0$	$) \wedge$																																																																																	
$(\langle 0,1 \rangle \rightarrow \langle 0,0 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 = 0 \wedge K_y^1 < 1$	$) \wedge$																																																																																	
$(\langle 0,1 \rangle \rightarrow \langle 0,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 = 0 \wedge K_y^1 = 1$	$) \wedge$																																																																																	
$(\langle 0,1 \rangle \rightarrow \langle 0,2 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 > 0 \wedge K_y^1 > 1$	$) \wedge$																																																																																	
$(\langle 0,1 \rangle \rightarrow \langle 1,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 > 0$	$) \wedge$																																																																																	
$(\langle 0,2 \rangle \rightarrow \langle 0,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 > 0 \wedge K_y^2 < 2$	$) \wedge$																																																																																	
$(\langle 0,2 \rangle \rightarrow \langle 0,2 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 > 0 \wedge K_y^2 = 2$	$) \wedge$																																																																																	
$(\langle 0,2 \rangle \rightarrow \langle 1,2 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 > 0$	$) \wedge$																																																																																	
$(\langle 1,0 \rangle \rightarrow \langle 0,0 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^1 < 1$	$) \wedge$																																																																																	
$(\langle 1,0 \rangle \rightarrow \langle 1,0 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^1 = 1 \wedge K_y^3 = 0$	$) \wedge$																																																																																	
$(\langle 1,0 \rangle \rightarrow \langle 1,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^1 = 1 \wedge K_y^3 > 0$	$) \wedge$																																																																																	
$(\langle 1,1 \rangle \rightarrow \langle 0,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 < 1$	$) \wedge$																																																																																	
$(\langle 1,1 \rangle \rightarrow \langle 1,0 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 < 1$	$) \wedge$																																																																																	
$(\langle 1,1 \rangle \rightarrow \langle 1,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 = 1 \wedge K_y^3 = 1$	$) \wedge$																																																																																	
$(\langle 1,1 \rangle \rightarrow \langle 1,2 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 = 1 \wedge K_y^3 > 1$	$) \wedge$																																																																																	
$(\langle 1,2 \rangle \rightarrow \langle 0,2 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 < 1$	$) \wedge$																																																																																	
$(\langle 1,2 \rangle \rightarrow \langle 1,1 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 < 1$	$) \wedge$																																																																																	
$(\langle 1,2 \rangle \rightarrow \langle 1,2 \rangle \in \mathcal{T} \Leftrightarrow$	$K_x^2 = 1 \wedge K_y^4 = 2$	$) \wedge$																																																																																	
"T ne peut contenir que les transitions apparaissant dans les équivalences ci-dessus"																																																																																			

 TAB. 3.1 – Illustration sur un exemple des notions fondamentales permettant de définir un modèle \mathcal{M} à partir d'un graphe \mathcal{G} d'interactions, et exemple de relation $r^{\mathcal{M}}(\mathcal{T}, \mathcal{P})$

les deux sous-ensembles d'interactions $\{\}$ et $\{i_x^1\}$ correspondant aux deux vecteurs V suivant : (0) et (1).

Les conditions $Cellc_{c,S}^l$ de contextes cellulaires pour ces sous-ensembles d'interactions et vecteurs V sont données dans la même Table 3.1. Nous donnons aussi dans cette figure la représentation graphique de ces contextes cellulaires. Par exemple, le contexte cellulaire associé à la condition $Cellc_{x,S}^1$ est composé des deux états $\langle 0, 0 \rangle$ et $\langle 1, 0 \rangle$, tandis que celui associé à la condition $Cellc_{x,S}^2$ est composé des quatre états $\langle 0, 1 \rangle$, $\langle 1, 1 \rangle$, $\langle 0, 2 \rangle$ et $\langle 1, 2 \rangle$.

Notons finalement que, par construction, une seule condition $Cellc_{c,S}^l$ est vraie pour un état S donné. De plus, les l_c régions définies par les conditions $Cellc_{c,S}^l$ constituent une partition de l'espace des concentrations discrètes.

3.1.3 Notion de composante focale

Dans les réseaux de Thomas à chaque état et chaque composante de cet état est associé une tendance, dite *composante focale*. Une composante focale, donnant la direction d'une composante c dans un état S donné, est dépendante du contexte cellulaire de c auquel S appartient. Afin de faire le lien entre contexte cellulaire et composante focale de manière formelle nous introduisons pour chaque contexte cellulaire un paramètre dit *paramètre cinétique discret*.

Définition 3.4 Soit c une composante. Un **paramètre cinétique discret** pour l'espèce c est une variable entière comprise entre 0 et \max_c .

Un tel paramètre est associé à un unique contexte cellulaire $Cellc_{c,S}^l$ relatif à c .

Nous identifions un tel paramètre par K_c^l . L'ensemble de ces paramètres K_c^l pour toutes les espèces constitue l'ensemble \mathcal{P} des paramètres du modèle.

Deux autres types de paramètres sont introduits pour généraliser la formalisation au Chapitre 5.

Exemple 3.4 Pour l'exemple de la Table 3.1, nous obtenons $l_x = 2$ paramètres pour la composante x : K_x^1 associé à la condition $Cellc_{x,S}^1$ et K_x^2 associé à la condition $Cellc_{x,S}^2$. Pour la composante y nous obtenons $l_y = 4$ paramètres K_y^l , chacun d'eux associé à une condition $Cellc_{y,S}^l$.

Nous pouvons alors définir la composante focale de c dans S par la Définition 3.5.

Définition 3.5 Soit c une composante. Soit S un état. La **composante focale** de c dans l'état S , notée $F_{c,S}$, est définie par l'**équation focale** de c suivante :

$$F_{c,S} = \sum_{l=1}^{l_c} K_c^l \cdot Cellc_{c,S}^l$$

Exemple 3.5 Les équations focales de x et y sont données dans la Table 3.1. Pour l'état $\langle 0, 0 \rangle$ la composante focale de x est $F_{x,\langle 0,0 \rangle} = K_x^1$, tandis que la composante focale de y est $F_{y,\langle 0,0 \rangle} = K_y^1$. Pour l'état $\langle 1, 1 \rangle$ nous obtenons $F_{x,\langle 1,1 \rangle} = K_x^2$ et $F_{y,\langle 1,1 \rangle} = K_y^3$.

Il est à noter que $F_{c,S}$ est obligatoirement égal à un des paramètres K_c^l : celui pour lequel $Cellc_{c,S}^l$ est vraie dans l'état S (en effet exactement une condition $Cellc_{c,S}^l$ est vraie pour tout état S).

3.1.4 Notion d'état focal

En Sous-Section 3.1.3 nous discutons de tendance d'une composante. Dans cette sous-Section nous considérons le système avec toutes ses composantes pour définir la tendance globale du système dans un état donné. C'est un état dit *état focal*.

Définition 3.6 Soit S un état. L'*état focal* de S , noté F_S est l'état composé de toutes les composantes focales de l'état S :

$$F_S = \langle F_{c_1,S}, F_{c_2,S}, \dots, F_{c_n,S} \rangle$$

Exemple 3.6 L'état focal de $\langle 0, 0 \rangle$ est $F_{\langle 0,0 \rangle} = \langle K_x^1, K_y^1 \rangle$. L'état focal de $\langle 1, 1 \rangle$ est $F_{\langle 1,1 \rangle} = \langle K_x^2, K_y^3 \rangle$.

L'état focal donne la direction de toutes les transitions débutant dans S . Le lien formel entre état focal et comportement dynamique est présenté en Section 3.2.

Finalement un modèle \mathcal{M} est précisé par la Définition 3.7.

Définition 3.7 Un *modèle* \mathcal{M} est un couple composé d'un système d'équations focales et d'un ensemble de contraintes sur les ensembles \mathcal{T} et \mathcal{P} associés à \mathcal{M} .

Il est à noter que pour l'instant nous pouvons uniquement ajouter à un modèle des contraintes exprimées sur l'ensemble \mathcal{P} . La section suivante va permettre de faire le lien formel entre \mathcal{T} et \mathcal{P} pour un système d'équation focales d'un modèle \mathcal{M} . Il sera alors possible d'exprimer aussi des contraintes sur l'ensemble \mathcal{T} .

3.2 Définition d'une relation entre l'ensemble \mathcal{T} des transitions et l'ensemble \mathcal{P} des paramètres

Rappelons que la relation $r^{\mathcal{M}}(\mathcal{T}, \mathcal{P})$ (présentée dans la Sous-Sous-Section 1.2.1) est vraie si \mathcal{T} est l'ensemble de transitions du modèle \mathcal{M} avec comme ensemble de paramètres \mathcal{P} suivant le formalisme de Thomas. Cette relation est définie informellement dans la Sous-Section 3.2.1 après avoir au préalable détaillé les conditions d'existence d'une transition dans le formalisme de Thomas.

Après discussion sur la représentation de cette relation en Sous-Section 3.2.2 il apparaît judicieux de définir plutôt une relation $transition(\mathcal{M}, t(S, S'), \mathcal{P})$ vraie si la transition de S vers S' existe dans l'ensemble de transitions \mathcal{T} pour le modèle \mathcal{M} ayant comme ensemble de paramètres \mathcal{P} .

La définition de cette dernière relation est donnée sous forme de contraintes dans la Sous-Section 3.2.3.

3.2.1 Définition informelle de $r^{\mathcal{M}}(\mathcal{T}, \mathcal{P})$

Suivant le formalisme de Thomas l'ensemble \mathcal{T} de transitions d'un modèle \mathcal{M} ayant pour ensemble de paramètres \mathcal{P} contient toutes les transitions $S \rightarrow S'$ respectant les conditions suivantes :

1. S' et S sont des états voisins ou égaux,

2. S' et S diffèrent d'une composante au plus¹⁰,
3. S' est dans la "direction" de l'état focal F_S de S .

Plus formellement, le troisième point conditionne l'existence dans \mathcal{T} des transitions respectant les deux premiers points de telle manière que :

- si $S = S'$ alors $F_S = S$ ("direction nulle"),
- si $S'_c = S_c + 1$ alors $F_{c,S} > S_c$ ("direction positive en c "),
- si $S'_c = S_c - 1$ alors $F_{c,S} < S_c$ ("direction négative en c ").

Exemple 3.7 Pour le modèle donné dans la Table 3.1 l'état $\langle 0, 0 \rangle$ a trois voisins : $\langle 0, 1 \rangle$, $\langle 1, 0 \rangle$ et $\langle 1, 1 \rangle$. Ainsi, les transitions respectant le premier point et débutant en $\langle 0, 0 \rangle$ sont : $\langle 0, 0 \rangle \rightarrow \langle 0, 1 \rangle$, $\langle 0, 0 \rangle \rightarrow \langle 1, 0 \rangle$, $\langle 0, 0 \rangle \rightarrow \langle 1, 1 \rangle$, $\langle 0, 0 \rangle \rightarrow \langle 0, 0 \rangle$. Le second point rejette la transition $\langle 0, 0 \rangle \rightarrow \langle 1, 1 \rangle$. Ainsi, il peut exister dans \mathcal{T} trois transitions débutant en $\langle 0, 0 \rangle$ dont l'existence dans \mathcal{T} est conditionné par le troisième point de la manière suivante :

- $\langle 0, 0 \rangle \rightarrow \langle 0, 1 \rangle$ est dans \mathcal{T} si l'état $\langle 0, 1 \rangle$ est dans la direction de $F_{\langle 0, 0 \rangle} = \langle K_x^1, K_y^1 \rangle$ (c'est-à-dire si $K_y^1 > \langle 0, 0 \rangle_y = 0$).
- $\langle 0, 0 \rangle \rightarrow \langle 1, 0 \rangle$ est dans \mathcal{T} si l'état $\langle 1, 0 \rangle$ est dans la direction de $F_{\langle 0, 0 \rangle} = \langle K_x^1, K_y^1 \rangle$ (c'est-à-dire si $K_x^1 > 0$).
- $\langle 0, 0 \rangle \rightarrow \langle 0, 0 \rangle$ est dans \mathcal{T} si l'état $\langle 0, 0 \rangle$ est stationnaire (c'est-à-dire si $F_{\langle 0, 0 \rangle} = \langle K_x^1, K_y^1 \rangle = \langle 0, 0 \rangle$).

L'ensemble de toutes les transitions respectant les deux premiers points et de leurs conditions d'existence dans \mathcal{T} associées est donné dans la dernière ligne de la Table 3.1.

Finalement la relation $r^{\mathcal{M}}(\mathcal{T}, \mathcal{P})$ est vraie si \mathcal{T} contient toutes les transitions respectant les trois points donnés ci-dessus pour le modèle \mathcal{M} ayant comme ensemble de paramètres \mathcal{P} .

3.2.2 Représentation informatique d'une relation formelle entre l'ensemble \mathcal{T} des transitions et l'ensemble \mathcal{P} des paramètres

Nous souhaitons trouver une bonne représentation informatique pour lier formellement \mathcal{T} et \mathcal{P} . Nous discutons pour cela des problèmes d'explosion de la taille des types de représentations déjà vu dans l'Exemple 1.1 et dans la Table 3.1. Finalement une nouvelle relation formelle entre \mathcal{T} et \mathcal{P} est proposée. Cette dernière permet d'éviter cette explosion de la taille de la représentation tout en étant suffisante pour exprimer l'existence d'une transition dans \mathcal{T} .

Une représentation comme une disjonction de valuations de \mathcal{P} et \mathcal{T} (voir l'Exemple 1.1) est lourde. En effet si nous traitons un problème avec l_{c_i} paramètres $K_{c_i}^l$ pour chaque espèce c_i dont les valeurs varient entre 0 et max_{c_i} l'espace des valeurs de paramètres contient $\prod_{i \in [1, n]} \prod_{l \in [1, l_{c_i}]} (max_{c_i} + 1)$ valuations.

Une représentation comme un ensemble de transitions entre états S et S' instanciés, et conditionnées par des propriétés sur \mathcal{P} (voir la dernière ligne de la Table 3.1) est aussi très lourde. En effet si nous traitons un problème avec n composantes c_i dont les concentrations varient entre 0 et max_{c_i} l'espace des concentrations contient $\prod_{i \in [1, n]} (max_{c_i} + 1)$ états.

Il apparaît alors judicieux de considérer une représentation similaire à la dernière discutée mais considérant une unique transition $S \rightarrow S'$ conditionnée par une propriété faisant intervenir

¹⁰Cette propriété d'asynchronicité est due au fait qu'il semble rare que les trajectoires continues passent plusieurs seuils exactement au même instant.

\mathcal{P} , S et S' (S et S' n'étant alors pas instanciés). La relation associée n'est alors plus $r^{\mathcal{M}}(\mathcal{T}, \mathcal{P})$ mais la relation $transition(\mathcal{M}, t(S, S'), \mathcal{P})$ vraie si la transition de S vers S' existe dans l'ensemble de transitions \mathcal{T} pour le modèle \mathcal{M} ayant comme ensemble de paramètres \mathcal{P} . Cette relation lie bien formellement \mathcal{T} et \mathcal{P} et ceci en évitant une explosion évidente de la taille de la représentation informatique associée mais aussi en étant suffisante pour exprimer l'existence d'une transition dans \mathcal{T} .

3.2.3 Définition formelle de $transition(\mathcal{M}, t(S, S'), \mathcal{P})$

Dans cette Sous-Section nous présentons le cœur du programme par contraintes comme un ensemble de prédicats (permettant de poser certaines contraintes) qui construit la relation ou prédicat $transition(\mathcal{M}, t(S, S'), \mathcal{P})$ entre une transition de l'ensemble \mathcal{T} des transitions et l'ensemble \mathcal{P} des paramètres cinétiques discrets d'un modèle \mathcal{M} .

Contraintes de transition.

La définition en contraintes de $transition(\mathcal{M}, t(S, S'), \mathcal{P})$ est donnée en Contrainte 3.1. Nous pouvons remarquer qu'elle est définie à l'aide de deux prédicats, d'une part $focal_state(\mathcal{M}, \mathcal{P}, S, F_S)$ vrai si et seulement si F_S est l'état focal de S pour le modèle \mathcal{M} considérant les paramètres \mathcal{P} , et d'autre part $transition_rule(t(S, S'), F_S)$ vrai si et seulement si la transition $S \rightarrow S'$ respecte les trois conditions d'une transition dans le formalisme de Thomas (comme donné en Sous-Section 3.2.1) avec F_S l'état focal de S^{11} .

Contrainte 3.1

$$\begin{aligned} transition(\mathcal{M}, t(S, S'), \mathcal{P}) &\Leftrightarrow \\ &focal_state(\mathcal{M}, \mathcal{P}, S, F_S) \wedge \\ &transition_rule(t(S, S'), F_S) \end{aligned}$$

Exemple 3.8 *Considérons le modèle \mathcal{M} présenté dans la Table 3.1 avec les contraintes supplémentaires $K_x^1 = 0$, $K_x^2 = 1$, $K_y^1 = 1$, $K_y^2 = 1$, $K_y^3 = 1$, $K_y^4 = 2$ que nous dénoterons par \mathcal{M}^C . L'ensemble \mathcal{T} de transitions pour cette valuation de \mathcal{P} est représenté par le graphe de transitions de la Figure 4.2 associé au modèle \mathcal{M}^C (dans le prochain chapitre).*

La requête $transition(\mathcal{M}^C, t(\langle 1, 2 \rangle, \langle 0, 2 \rangle), \mathcal{P})$ répond positivement signifiant que l'ensemble des contraintes posées est cohérent. On peut vérifier cela en constatant l'existence de la transition $\langle 1, 2 \rangle \rightarrow \langle 0, 2 \rangle$ dans le graphe de transition \mathcal{T} correspondant à \mathcal{M}^C de la Figure 4.2.

Exemple 3.9 *Par contre, toujours pour le modèle \mathcal{M}^C (comme défini dans l'Exemple 3.8) la requête $transition(\mathcal{M}^C, t(\langle 0, 1 \rangle, \langle 1, 1 \rangle), \mathcal{P})$ répond négativement, indiquant une incohérence. Cela provient du fait que \mathcal{T} n'accepte pas la transition $\langle 0, 1 \rangle \rightarrow \langle 1, 1 \rangle$. Nous verrons d'où provient cette incohérence dans les exemples au fil des descriptions formelles de cette section.*

Contraintes d'état focal.

$focal_state(\mathcal{M}, \mathcal{P}, S, F_S)$ est vrai si le système de contraintes fondé par la duplication de l'information contenue dans les équations focales de \mathcal{M} est consistant. La Contrainte 3.2 donne la définition formelle de ce prédicat où les $Cellc_{c,S}^l$ sont des variables Booléennes et les K_c^l sont des variables entières contraintes d'après les équations focales du modèle \mathcal{M} .

¹¹Nous remarquons bien ici comment la notion intermédiaire d'état focal permet de faire le lien entre un modèle \mathcal{M} et ses paramètres \mathcal{P} d'une part et une transition de \mathcal{T} d'autre part.

Contrainte 3.2

$$\text{focal_state}(\mathcal{M}, \mathcal{P}, S, F_S) \Leftrightarrow \bigwedge_{l,c} \text{Cell}_{c,S}^l \Rightarrow F_{c,S} = K_c^l$$

Exemple 3.10 *Considérons le modèle \mathcal{M} donné dans la Table 3.1. La requête $\text{focal_state}(\mathcal{M}, \mathcal{P}, S, F_S) \wedge (F_S = [0, H]) \wedge (S = [G, 2])$ (où G et H sont de nouvelles variables entières) est équivalente à la cohérence des équations obtenues en substituant les variables F_S , \mathcal{P} et S aux variables correspondantes dans les équations focales de \mathcal{M} comme montré dans la Figure 3.1(a). La substitution conduit au système de contraintes donné en Figure 3.1(b) qui est simplifié en Figure 3.1(c). Ce système étant cohérent nous recevons une réponse positive.*

$F_{x,S} \rightsquigarrow 0$
$F_{y,S} \rightsquigarrow H$
$K_c^j \rightsquigarrow K_c^j$
$S_x \rightsquigarrow G$
$S_y \rightsquigarrow 2$
$t_x^1 \rightsquigarrow 1$
$t_y^1 \rightsquigarrow 1$
$t_y^2 \rightsquigarrow 2$

$$\left\{ \begin{array}{l} 0 = \begin{array}{l} K_x^1 \cdot (2 < 1) \\ + K_x^2 \cdot (2 \geq 1) \end{array} \\ \\ H = \begin{array}{l} K_y^1 \cdot (G < 1) \wedge (2 < 2) \\ + K_y^2 \cdot (G < 1) \wedge (2 \geq 2) \\ + K_y^3 \cdot (G \geq 1) \wedge (2 < 2) \\ + K_y^4 \cdot (G \geq 1) \wedge (2 \geq 2) \end{array} \end{array} \right.$$

(a) (b)

$$\left\{ \begin{array}{l} 0 = K_x^2 \\ \\ H = \begin{array}{l} K_y^2 \cdot (G < 1) \\ + K_y^4 \cdot (G \geq 1) \end{array} \end{array} \right.$$

(c)

FIG. 3.1 – Exemple de création de contraintes équivalentes à “ $\langle 0, H \rangle$ est l’état focal de $\langle G, 2 \rangle$ pour le modèle \mathcal{M} de la Table 3.1”

Exemple 3.11 *Considérons le modèle \mathcal{M}^C défini dans l’Exemple 3.8. Pour ce modèle nous avons $K_x^2 = 0$ et $K_y^1 = 1$. La requête $\text{focal_state}(\mathcal{M}^C, \mathcal{P}, \langle 0, 1 \rangle, F_S)$ est équivalente à la contrainte $F_S = \langle K_x^2, K_y^1 \rangle = \langle 0, 1 \rangle$ et reçoit donc une réponse positive.*

Contraintes de règle de transition.

Le prédicat $\text{transition_rule}(t(S, S'), F_S)$ est défini par la Contrainte 3.3 où la notation o correspond à la concaténation de listes et $\text{sum}(L, Sum)$ est vrai si Sum est la somme des entiers (ici uniquement des Booléens) de la liste L .

Contrainte 3.3

$$\begin{array}{l}
 \text{transition_rule}(t(S, S'), F_S) \Leftrightarrow \\
 \bigwedge_c (LBup_c \Leftrightarrow S'_c = S_c + 1) \wedge \\
 \bigwedge_c (LBdown_c \Leftrightarrow S'_c = S_c - 1) \wedge \\
 (Bst \Leftrightarrow S = S') \wedge \\
 \left. \begin{array}{l}
 \bigwedge_c (S_c - 1 \leq S'_c \leq S_c + 1) \wedge \\
 \text{sum}(LBup \circ LBdown \circ [Bst], 1) \wedge
 \end{array} \right\} \begin{array}{l} \text{point 1} \\ \text{point 2} \end{array} \\
 \left. \begin{array}{l}
 \bigwedge_c (LBup_c \Rightarrow S_c < F_{c,S}) \wedge \\
 \bigwedge_c (LBdown_c \Rightarrow S_c > F_{c,S}) \wedge \\
 (Bst \Leftrightarrow S = F_S)
 \end{array} \right\} \text{point 3}
 \end{array}$$

Ce prédicat formalise et implémente la propriété d'appartenance d'une transition à \mathcal{T} selon le formalisme de Thomas en trois points comme précédemment vu dans la Sous-Section 3.2.1. Chacun de ces points est formalisé par un ensemble de contraintes identifié dans Contrainte 3.3 par une accolade associée.

Différentes variables intermédiaires apparaissent afin de factoriser les expressions communes ($S'_c = S_c + 1$ par exemple) à la formalisation des points 2 et 3. $LBup$ et $LBdown$ sont des listes de variables Booléennes, et Bst une variable Booléenne. $LBup_c$ et $LBdown_c$ dénotent les variables des listes $LBup$ et $LBdown$ d'indice c . Chacune de ces variables est associée à une des $2 * n + 1$ directions possibles (n étant, nous le rappelons, le nombre d'espèces du modèle \mathcal{M} de S' par rapport à S). $LBup_c$, respectivement $LBdown_c$ est vrai si et seulement si la transition $S \rightarrow S'$ fait croître, respectivement décroître, la composante c de 1. Bst est vrai si et seulement si la transition $S \rightarrow S'$ est stationnaire.

Exemple 3.12 La requête $\text{transition_rule}(t(\langle 0, 0 \rangle, S'), \langle 1, 1 \rangle)$ conduit à un succès et fournit deux solutionsinstanciées : $S' = \langle 1, 0 \rangle$ et $S' = \langle 0, 1 \rangle$. Ces deux états S' sont les successeurs possibles de $\langle 0, 0 \rangle$.

Les contraintes interviennent ainsi : S' ne peut être égal à $\langle 1, 1 \rangle$ et $\langle 0, 0 \rangle$ à cause des deux contraintes $Bst \Leftrightarrow S = S'$ et $Bst \Leftrightarrow S = F_S$, S' ne peut être égal à $\langle 0, 2 \rangle$ et $\langle 1, 2 \rangle$ à cause des contraintes associé au point 1 ($\bigwedge_c (S_c - 1 \leq S'_c \leq S_c + 1)$). Les deux états possibles restants sont $\langle 1, 0 \rangle$ et $\langle 0, 1 \rangle$. Ils satisfont la requête.

Exemple 3.13 La requête $\text{transition_rule}(t(\langle 0, 1 \rangle, \langle 1, 1 \rangle), \langle 0, 1 \rangle)$ conduit à un échec car Bst serait faux du fait que $(Bst \Leftrightarrow S = S')$ avec $S = \langle 0, 1 \rangle \neq \langle 1, 1 \rangle = S'$ et Bst serait faux du fait que $(Bst \Leftrightarrow S = F_S)$ avec $S = \langle 0, 1 \rangle$ et $F_S = \langle 0, 1 \rangle$.

Finalement, nous pouvons expliquer pourquoi la requête de l'Exemple 3.9, $\text{transition}(\mathcal{M}, t(\langle 0, 1 \rangle, \langle 1, 1 \rangle) \wedge \mathcal{P})$, conduit à un échec. D'après la définition du prédicat transition cette requête est vraie si et seulement si $\text{focal_state}(\mathcal{M}, \mathcal{P}, \langle 0, 1 \rangle, F_S) \wedge \text{transition_rule}(t(\langle 0, 1 \rangle, \langle 1, 1 \rangle), F_S)$ est vrai. Or, nous avons vu que la requête de l'Exemple 3.11, $\text{focal_state}(\mathcal{M}, \mathcal{P}, \langle 0, 1 \rangle, F_S)$, conduit à la contrainte $F_S = \langle 0, 1 \rangle$, et nous avons vu ci-dessus que la requête $\text{transition_rule}(t(\langle 0, 1 \rangle, \langle 1, 1 \rangle), F_S) \wedge F_S = \langle 0, 1 \rangle$ conduit à un échec. Donc $\text{focal_state}(\mathcal{M}, \mathcal{P}, \langle 0, 1 \rangle, F_S) \wedge \text{transition_rule}(t(\langle 0, 1 \rangle, \langle 1, 1 \rangle), F_S)$ conduit à un échec.

Chapitre 4

Définition de propriétés de réseaux de Thomas

Nous proposons dans ce chapitre de définir deux types de propriétés associables à des réseaux de Thomas. Le premier concerne des propriétés sur la manière de composer des interactions, tandis que le deuxième concerne des propriétés sur les comportements dynamiques. Ces propriétés sont destinées soit à être imposées, soit à être vérifiées ou à être inférées depuis un modèle.

4.1 Propriétés sur la manière de composer des interactions

Rappelons que le graphe d'interaction \mathcal{G} fournit un ensemble d'interactions possibles sur chacun de ses gènes. Mais il n'indique pas la manière de composer ces interactions. Par contre une valuation spécifique de l'ensemble des paramètres K_c^l décrit précisément la direction de la composante c , et offre donc une information précise sur la manière dont se compose les interactions sur c .

Exemple 4.1 *Considérons le modèle \mathcal{M} donné dans la Table 3.1. Soient les deux interactions sur l'espèce y $i_y^1 = (x, t_x^1, y)$ et $i_y^2 = (y, t_y^2, y)$. Une valuation des quatre paramètres K_y^l , $l \in [1, 4]$, décrit précisément la manière dont sont composées i_y^1 et i_y^2 . Soit par exemple la valuation $\{K_y^1 = 0, K_y^2 = 0, K_y^3 = 0, K_y^4 = 2\}$. Elle indique précisément que la composante focale y est égale à 2 pour un état S appartenant au contexte cellulaire défini par les conditions $I_{y,S}^1 = 1$ et $I_{y,S}^2 = 1$, et qu'elle est nulle si $I_{y,S}^1 = 0$ ou $I_{y,S}^2 = 0$.*

En tenant compte uniquement des groupes de valeurs de composantes focales nous pouvons dire que :

- les composantes focales y des états S où $I_{y,S}^1 \wedge I_{y,S}^2$ sont toutes égales à une valeur $F1$,
- les composantes focales y des états S où $\neg(I_{y,S}^1 \wedge I_{y,S}^2)$ sont toutes égales à une valeur $F0$.

L'espace des concentrations discrètes est donc partitionné en deux régions identifiées par la valeur de $I_{y,S}^1 \wedge I_{y,S}^2$, ou en inversant les valeurs par $\neg(I_{y,S}^1) \vee \neg(I_{y,S}^2)$. Une interprétation possible de la manière de composer les interactions i_y^1 et i_y^2 est alors la “conjonction” des effets de i_y^1 et i_y^2 , ou alors la “disjonction” des effets “inverse” de i_y^1 et i_y^2 .

En ajoutant la connaissance sur l'ordre entre $F0$ et $F1$ (ici $F0 < F1$) nous pouvons choisir préférentiellement la première interprétation (“conjonction” des effets de i_y^1 et i_y^2).

Au vue de la formalisation présentée dans le Chapitre 3 il est possible d'imposer et d'inférer des propriétés concernant les manières de composer les interactions en termes des paramètres K_c^l . Cependant, l'interprétation à la main de valuations ou de propriétés sur les paramètres K_c^l n'est pas aisée, en particulier à un utilisateur peu familier avec la formalisation des réseaux de Thomas. Il s'agit donc de concevoir un langage de plus haut niveau pour imposer, vérifier et inférer des propriétés partielles concernant la manière de composer les interactions de \mathcal{G} . Ce langage est nommé *LComp*.

Dans un premier temps nous considérons des propriétés sur la manière de composer les interactions pour obtenir un découpage plus fin de l'espace des concentrations discrètes (voir le deuxième paragraphe de l'Exemple 4.1). La syntaxe et la sémantique de cette première version de *LComp* sont présentées en Sous-Section 4.1.1.

Dans un deuxième temps nous considérons deux nouvelles propriétés sur la manière de composer les interactions, en précisant l'ordre des composantes focales pour les différentes régions de l'espace des concentrations discrètes (voir le dernier paragraphe de l'Exemple 4.1). Cette deuxième version de *LComp*, présentée en Sous-Sections 4.1.2 et 4.1.3, étend la première.

L'intérêt de séparer syntaxiquement et sémantiquement ces différents niveaux de précision sur la manière de composer les interactions est de pouvoir choisir finement ce qui doit être imposé, vérifié ou inféré, tout en restant au niveau de compréhension d'un utilisateur peu familier avec la formalisation des réseaux de Thomas.

4.1.1 Le langage *LComp* d'expression de compositions d'interactions

Nous décrivons dans cette sous-section la syntaxe et la sémantique de la première version du langage *LComp*. Les deux premières sous-sous sections suivantes étendent et font référence aux notions présentées en Sous-Sections 3.1.1 et 3.1.2, tandis que la troisième sous-sous-section fait référence à la Sous-Section 3.1.3.

Notion de composition d'interactions et d'activité d'une composition d'interactions

Chaque manière de composer les interactions sur un gène donné est appelée une *composition d'interactions*. Le langage *LComp* permet de caractériser un ensemble de compositions d'interactions pour une espèce donnée. Chaque composition d'interaction est exprimée selon la grammaire suivante :

$$\begin{aligned} \textit{Composition} & ::= \textit{Interaction} \mid (\dot{-} \textit{Interaction}) \mid \\ & \quad (\textit{Composition} \dot{\wedge} \textit{Composition}) \mid (\textit{Composition} \dot{\vee} \textit{Composition}) \\ \textit{Interaction} & ::= i_c^r \end{aligned}$$

Exemple 4.2 Pour l'exemple donné dans la Table 3.1 l'espèce y possède deux interactions i_y^1 et i_y^2 . Avec ces deux interactions nous pouvons exprimer, par exemple, les compositions suivantes : i_y^1 , $(\dot{-}i_y^2)$, $(i_y^1 \dot{\wedge} i_y^2)$, $((\dot{-}i_y^1) \dot{\vee} (\dot{-}i_y^2))$ et $(((\dot{-}i_y^1) \dot{\wedge} i_y^2) \dot{\vee} (i_y^1 \dot{\wedge} (\dot{-}i_y^2)))$.

L'activité d'une composition d'interactions dans un état S est définie à partir de celle d'une interaction (voir Sous-Section 3.1.1).

Définition 4.1 L'activité d'une composition d'interactions J sur l'espèce c dans l'état S est :

- $I_{c,S}^r$ si $J = i_c^r$,
- $\neg I_{c,S}^r$ si $J = (\dot{-}i_c^r)$,
- $J1 \wedge J2$ si $J = (I1 \dot{\wedge} I2)$,

– $J1 \vee J2$ si $J = (I1 \dot{\vee} I2)$,
où $J1$ est l'activité de $I1$ dans l'état S et $J2$ celle de $I2$ dans S .

Exemple 4.3 Pour l'exemple de la Table 3.1, la composition d'interactions $(i_y^1 \wedge i_y^2)$ est active pour les états S tel que $(I_{y,S}^1 \wedge I_{y,S}^2)$, donc pour l'état $\langle 1, 2 \rangle$. De la même manière, la composition $(i_y^1 \dot{\vee} i_y^2)$ est active pour les états $\langle 0, 2 \rangle$, $\langle 1, 0 \rangle$, $\langle 1, 1 \rangle$ et $\langle 1, 2 \rangle$. La composition $(i_y^1 \wedge (\dot{-}i_y^2))$ est active pour les états $\langle 1, 0 \rangle$ et $\langle 1, 1 \rangle$.

Soit un ensemble E_c de q_c compositions d'interactions définies pour la cibles c . Nous associons un indice q , $1 \leq q \leq q_c$, à chacune de ses compositions. Nous représentons un ensemble E_c par un ensemble de q_c couples (J, q) où J est une composition d'interaction sur c et q son indice. L'activité d'une composition d'interactions d'indice q de E_c dans l'état S est notée $J_{c,S}^q$.

Exemple 4.4 Pour l'exemple donné dans la Table 3.1 nous avons l'unique interaction $i_x^1 = (y, t_y^1, x)$ sur x , et les deux interactions $i_y^1 = (x, t_x^1, y)$ et $i_y^2 = (y, t_y^2, y)$ sur y . Considérons les ensembles de compositions d'interactions suivants :

$$E_x = \{(\dot{-}i_x^1, 1)\} \text{ pour } x,$$

$$E_y = \{(i_y^1, 1), (i_y^2, 2)\} \text{ pour } y.$$

L'activité de la composition d'interaction $(\dot{-}i_x^1)$ pour l'espèce x dans l'état S , notée $J_{x,S}^1$ ¹², est équivalente à $\neg I_{x,S}^1$. L'activité de i_y^1 d'indice 1 pour l'espèce y dans l'état S , noté $J_{y,S}^1$, est équivalente à $I_{y,S}^1$. L'activité de i_y^2 d'indice 2 pour l'espèce y dans l'état S , noté $J_{y,S}^2$, est équivalente à $I_{y,S}^2$.

Remarquons finalement que si l_c est le nombre de contextes cellulaires pour l'espèce c il est possible d'exprimer au maximum 2^{l_c} compositions d'interactions dont les activités ne sont pas équivalentes. Comme $l_c \leq 2^{r_c}$ il est impossible d'exprimer plus de $2^{(2^{r_c})}$ compositions d'interactions dont les activités ne sont pas équivalentes.

Notion de composition de contextes cellulaires

Un ensemble E_c de toutes les compositions d'interactions relatives à une espèce c , comme en Sous-Section 3.1.2, induit une partition de l'espace des concentrations. Chaque sous-ensembles de E_c définit une région (potentiellement vide) de l'espace des concentrations dite *composition de contextes cellulaires* qui est composée elle-même de contextes cellulaires. Comme en Sous-Section 3.1.2 un tel ensemble est caractérisé par une valuation d'un vecteur de Booléens, soit W , de taille q_c . Plus précisément :

Définition 4.2 Soit W un vecteur de Booléens de q_c éléments et m la valeur entière dans le système binaire que W représente. La **composition de contextes cellulaires** associée à W est composé des états S satisfaisant $Comp_{c,S}^m$ tel que :

$$Comp_{c,S}^m \equiv \bigwedge_{q=1}^{q_c} (W_q \Leftrightarrow J_{c,S}^q)$$

¹²Il est à noter que le 1 de $J_{x,S}^1$ correspond à l'indice de la composition d'interactions $(\dot{-}i_x^1)$ et non à l'indice de l'interaction i_x^1 .

Langage $LComp$

Les éléments du langage $LComp$ relatif à une composante c sont les ensembles E_c possibles. La sémantique d'un élément E_c est la suivante : pour toute valuation W , tous les états de la composition de contextes cellulaires associée à W ont la même composante focale c . En d'autres termes, les paramètres K_c^l associés à des contextes cellulaires appartenant à une même composition de contextes cellulaires sont égaux. Il est à noter que nous avons vu en Sous-Section 3.1.3 que cela est automatiquement vrai pour les contextes cellulaires : tous les état d'un même contexte cellulaire de c ont la même composante focale c .

Plus précisément, la contrainte relative à un ensemble E_c de compositions d'interactions est la suivante :

Contrainte 4.1

Pour toute composition de contextes cellulaires définie par $Comp_{c,S}^m$
 Soit L défini par $l \in L \Leftrightarrow \exists S | (Cellc_{c,S}^l \wedge Comp_{c,S}^m)$
 $\forall l, l' \in L, \quad K_c^l = K_c^{l'}$

L'ensemble L définit l'union des contextes cellulaires $Cellc_{c,S}^l$ formant la composition $Comp_{c,S}^m$ par l'intermédiaire des indices l des K_c^l et $Cellc_{c,S}^l$.

Exemple 4.5 Pour l'exemple de la Table 3.1 avec les hypothèses de compositions d'interactions $\{(\neg i_x^1, 1)\}$ pour x , et $\{(i_y^1 \wedge i_y^2), 1\}$ pour y , nous obtenons les sous-ensembles de compositions d'interactions, vecteurs W , compositions de contextes cellulaires $Comp_{c,S}^m$ (en terme de $J_{c,S}^q$ puis en terme de $I_{c,S}^r$), et ensembles L suivants :

sous-ensemble de compositions d'interactions	relatif à x		relatif à y	
	$\{\}$	$\{(\neg i_x^1)\}$	$\{\}$	$\{(i_y^1 \wedge i_y^2)\}$
vecteur W	$[0]$	$[1]$	$[0]$	$[1]$
$Comp_{c,S}^m$ (en terme des $J_{c,S}^q$)	$\neg J_{x,S}^1$	$J_{x,S}^1$	$\neg J_{y,S}^1$	$J_{y,S}^1$
$Comp_{c,S}^m$ (en terme des $I_{c,S}^r$)	$I_{x,S}^1$	$\neg I_{x,S}^1$	$\neg I_{y,S}^1 \vee \neg I_{y,S}^2$	$I_{y,S}^1 \wedge I_{y,S}^2$
ensemble L	$\{2\}$	$\{1\}$	$\{1, 2, 3\}$	$\{4\}$

Ainsi, nous obtenons les contraintes suivantes pour les hypothèses de compositions d'interactions : $K_y^1 = K_y^2 = K_y^3$.

Exemple 4.6 Toujours pour l'exemple de la Table 3.1 mais avec les hypothèses de compositions d'interactions suivantes : $\{\}$ pour x , et $\{((\neg i_y^1) \wedge i_y^2), 1\}$, $\{(i_y^1 \wedge (\neg i_y^2)), 2\}$ pour y , nous obtenons les sous-ensembles de compositions d'interactions, vecteurs W , compositions de contextes cellulaires $Comp_{c,S}^m$ (en terme de $J_{c,S}^q$ puis en terme de $I_{c,S}^r$), et ensembles L suivants :

sous-ensemble de compositions d'interactions	relatif à x	relatif à y			
	$\{\}$	$\{\}$	$\{(i_y^1 \wedge (\neg i_y^2))\}$	$\{((\neg i_y^1) \wedge i_y^2)\}$	$\{((\neg i_y^1) \wedge i_y^2), (i_y^1 \wedge \neg i_y^2)\}$
vecteur W	$[\]$	$[0, 0]$	$[0, 1]$	$[1, 0]$	$[1, 1]$
$Comp_{c,S}^m$	<i>true</i>	$\neg J_{y,S}^1 \wedge \neg J_{y,S}^2$	$\neg J_{y,S}^1 \wedge J_{y,S}^2$	$J_{y,S}^1 \wedge \neg J_{y,S}^2$	$J_{y,S}^1 \wedge J_{y,S}^2$
$Comp_{c,S}^m$	<i>true</i>	$(I_{y,S}^1 \vee \neg I_{y,S}^2) \wedge (\neg I_{y,S}^1 \vee I_{y,S}^2)$	$I_{y,S}^1 \wedge \neg I_{y,S}^2$	$\neg I_{y,S}^1 \wedge I_{y,S}^2$	<i>false</i>
ensemble L	$\{1, 2\}$	$\{1, 4\}$	$\{2\}$	$\{3\}$	$\{\}$

Ainsi, nous obtenons les contraintes suivantes : $K_x^1 = K_x^2$ et $K_y^1 = K_y^4$.

Exemple 4.7 Selon les hypothèses de compositions d'interactions données dans l'Exemple 4.4 nous n'avons aucune contraintes à ajouter étant donné que les ensembles L sont tous des singletons. En terme de régions d'états cela s'explique par le fait que les compositions d'interactions n'impliquent pas de regroupement de contextes cellulaires pour former les compositions de contextes cellulaires.

Précisons que l'utilisateur n'a à spécifier que les propriétés de compositions d'interactions qu'il souhaite. Le logiciel se charge de dériver les contraintes sur les paramètres correspondants. L'utilisateur garde la possibilité de ne pas utiliser ce langage et d'exprimer des contraintes directement sur les paramètres ou sur les comportements devant exister.

4.1.2 Additivité d'une composition d'interactions

Informellement, une composition d'interaction sur c est additive si son activité ne peut faire décroître la composante focale c .

L'introduction de cette propriété nous apparaît naturelle d'une part car elle apparaît systématiquement si l'on modélise un réseau génétique par un système d'équation différentielle (voir la note de bas de page associé à l'Exemple 4.4) faisant intervenir une *somme* de termes de production $\kappa_{c,i} * b$ où b est une variable Booléenne.

D'autre part nous souhaitons dans le cas d'un *reverse-engineering* de compositions d'interactions fournir à l'utilisateur des possibles compositions d'interactions additives étant donné que si elle ne le sont pas cela indique que nous pouvons trouver d'autres compositions qui sont cette fois additives et sont donc plus informatives étant donné qu'elles ont une propriété de plus.

Définition 4.3 Une composition d'interaction I sur c est additive, noté $a(I)$, si **pour tout** couple d'états $(S0, S1)$ tel que :

- $S0$ et $S1$ sont adjacents,
- I est non active dans $S0$,
- I est active dans $S1$,

la composante focale c dans $S0$ est inférieure ou égale à la composante focale c dans $S1$.

Afin de formaliser cette définition en terme des paramètres de \mathcal{P} , il faut faire le lien entre contextes cellulaires identifié par des conditions $Cellc_{c,S}^l$ qui sont chacune relative à un paramètre K_c^l (voir Section 3.1) et l'additivité d'une composition d'interactions. Pour cela introduisons deux définitions simples.

Définition 4.4 Deux contextes cellulaires identifiés par $Cellc_{c,S}^{l0}$ et $Cellc_{c,S}^{l1}$ sont adjacents si ils sont des régions non vide ($Cellc_{c,S}^{l0}$ et $Cellc_{c,S}^{l1}$ peuvent être vrai) et que leurs vecteurs Booléens V les caractérisant différent exactement d'un élément.

Définition 4.5 Une composition d'interaction I est active dans un contexte cellulaire identifié par la condition $Cellc_{c,S}^l$ si elle est active dans les états S tel que $Cellc_{c,S}^l$ est vrai.

La formalisation en terme des paramètres \mathcal{P} de l'additivité d'une composition d'interactions est alors la suivante (la démonstration est triviale) :

Contrainte 4.2

Une composition d'interaction I sur c est additive si et seulement si
pour tout couple de conditions $(Cellc_{c,S}^{l_0}, Cellc_{c,S}^{l_1})$ tel que

- $Cellc_{c,S}^{l_0}$ et $Cellc_{c,S}^{l_1}$ sont adjacents, et
- I est non active dans la région $Cellc_{c,S}^{l_0}$, et
- I est active dans la région $Cellc_{c,S}^{l_1}$,

on a $K_c^{l_0} \leq K_c^{l_1}$

Nous adoptons dans le langage $LComp$ un abus de langage pour l'ajout d'une propriété d'additivité $a(I)$ ajoutée à un ensemble de compositions d'interactions E_c noté naturellement $E_c \wedge a(I)$. Cet abus de langage consiste à remplacer le couple (I, q) de E_c par $(a(I), q)$.

Exemple 4.8 Aux hypothèses de compositions d'interactions données dans l'Exemple 4.4 nous ajoutons les contraintes relatives à leur additivité, soit $a((\dot{-}i_x^1))$, $a(i_y^1)$ et $a(i_y^2)$. Les hypothèses de compositions d'interactions deviennent $\{(a((\dot{-}i_x^1)), 1)\}$ pour x , et $\{(a(i_y^1), 1), (a(i_y^2), 2)\}$ pour y .

Pour l'unique composition d'interactions $I = (\dot{-}i_x^1)$ sur l'espèce x , I est non active dans la région définie par $Cellc_{x,S}^2$ et I est active dans la région définie par $Cellc_{x,S}^1$. Comme les conditions $Cellc_{x,S}^1$ et $Cellc_{x,S}^2$ sont les seules à considérer, l'additivité $a((\dot{-}i_x^1))$ est équivalente à la contrainte $K_x^2 \leq K_x^1$.

Pour la composition d'interactions (i_y^1) sur l'espèce y , deux couples de conditions $(Cellc_{y,S}^{l_0}, Cellc_{y,S}^{l_1})$ respectent la formalisation de l'additivité. L'un d'eux est tel que $l_0 = 1$ et $l_1 = 3$, et l'autre tel que $l_0 = 2$ et $l_1 = 4$. On peut pour s'en convaincre se référer à la Table 3.1 et en particulier à l'illustration graphique des contextes cellulaires (qui sont égaux dans cet exemple aux compositions de contextes cellulaires). Ainsi, l'additivité $a(i_y^1)$ est équivalente à $(K_y^1 \leq K_y^3) \wedge (K_y^2 \leq K_y^4)$.

De manière similaire l'additivité $a(i_y^2)$ est équivalente à $(K_y^1 \leq K_y^2) \wedge (K_y^3 \leq K_y^4)$.

Remarquons qu'une contrainte d'additivité d'une composition d'interactions sur c n'est pas dépendante de l'ensemble complet des compositions d'interactions sur c .

Ainsi il est possible d'ajouter une contrainte d'additivité d'une composition d'interactions sans connaître l'ensemble complet de compositions d'interactions. Par contre étant donné le langage il est nécessaire de spécifier l'ensemble complet de compositions d'interactions sur un gène c . L'astuce est alors d'ajouter à l'ensemble de compositions d'interactions supposées toutes les compositions réduites à des interactions. Ainsi il n'y a aucune réunion de régions associées aux contextes cellulaires lors de la prise en compte de l'ensemble de compositions d'interactions, et il n'y a donc aucune contrainte ajoutée par la Contrainte 4.1.

4.1.3 Observabilité d'une composition d'interactions

Informellement, une composition d'interactions I sur c est observable si l'effet de son activité est détectable dans \mathcal{T} . C'est-à-dire que nous trouvons dans \mathcal{T} une différence de comportement dynamique suivant la composante c à la "frontière" des espaces des concentrations définis par l'activité de I .

Nous introduisons cette propriété d'une part car elle permet au biologiste d'imposer facilement les contraintes relatives à l'observation d'une interaction (au sens général), et d'autre part car

nous souhaitons dans le cas d'un *reverse-engineering* de compositions d'interactions fournir à l'utilisateur des possibles compositions d'interactions observables. De la même manière que pour la propriété d'additivité il est préférable de fournir à l'utilisateur des possibles compositions d'interactions les plus informatives possibles et pour cela nous les souhaitons observables (et additives).

Définition 4.6 Une composition d'interactions I sur c , noté $o(I)$, est observable s'il existe un couple d'états $(S0, S1)$ tel que :

- $S0$ et $S1$ sont adjacents,
- I est non active dans $S0$,
- I est active dans $S1$,

et une des trois conditions suivantes est satisfaite :

1. $S0$ et $S1$ diffèrent sur une composante différente de c
 - soit il existe une transition partant de $S1$ augmentant suivant c , et il n'existe pas de transition partant de $S0$ augmentant suivant c ,
 - soit il existe une transition partant de $S0$ diminuant suivant c , et il n'existe pas de transition partant de $S1$ diminuant suivant c ,
2. $S0$ et $S1$ diffèrent sur la composante c , $S0_c < S1_c$, et les transitions $S0 \rightarrow S1$ et $S1 \rightarrow S0$ n'existent pas,
3. $S0$ et $S1$ diffèrent sur la composante c , $S0_c > S1_c$, et les transitions $S0 \rightarrow S1$ et $S1 \rightarrow S0$ existent.

Les trois conditions 1., 2. et 3. de la Définition 4.6 en terme de \mathcal{T} de l'observabilité d'une composition d'interactions sont illustrés par la Figure 4.1.

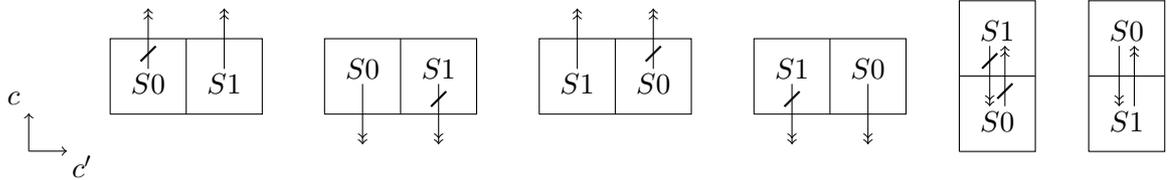


FIG. 4.1 – Représentation des 3 conditions de l'observabilité d'une composition d'interactions I sur c par un ensemble de 6 configurations de transitions, où $S0$ et $S1$ sont deux états adjacents où I n'est pas active dans $S0$ et I est active dans $S1$. La composante c est verticale ainsi que les transitions ou non autorisation de transitions car il s'agit de détecter un effet de la composition d'interactions I sur cette composante c . La composante c' est une composante différente de c quelconque. Les 4 configurations de gauche correspondent à la première condition de la Définition 4.6, la cinquième configuration à la deuxième condition et la sixième configuration à la troisième condition. Concernant la cinquième configuration il est à noter qu'*a priori* toutes les transitions partant de $S0$ et $S1$ sont autorisées sauf les deux transitions barrées dans la figure.

D'après cette définition de l'observabilité en terme de \mathcal{T} il s'agit de déduire une propriété équivalente en terme de \mathcal{P} . Pour cela nous devons faire le lien entre états et contextes cellulaires étant donné que la définition s'exprime en terme d'états et que pour exprimer une propriété en terme de \mathcal{P} il nous faut faire le lien avec les contextes cellulaires (les paramètres K_c^l de \mathcal{P} étant associé aux conditions $Cellc_{c,S}^l$). Il est nécessaire d'introduire les valeurs minimale et maximale

de la composante c acceptées par une condition $Cellc_{c,S}^l$ que nous dénotons respectivement par $min_{c,l}$ et $max_{c,l}$. La propriété en terme de \mathcal{P} équivalente à la Définition 4.6 de l'observabilité d'une composition d'interaction est donnée par la Contrainte 4.3. La démonstration de cette équivalence est donnée en Annexe C.

Contrainte 4.3

Une interaction complexe I sur c est observable, noté $o(I)$, si et seulement si il existe un couple de conditions $(Cellc_{c,S}^{l_0}, Cellc_{c,S}^{l_1})$ tel que

- $Cellc_{c,S}^{l_0}$ et $Cellc_{c,S}^{l_1}$ sont adjacentes, et
- I est non active dans la région $Cellc_{c,S}^{l_0}$, et
- I est active dans la région $Cellc_{c,S}^{l_1}$, et

- $K_c^{l_0} < K_c^{l_1}$, et
- $K_c^{l_0} \leq max_{c,l_0}$, et
- $K_c^{l_0} \leq max_{c,l_1}$, et
- $min_{c,l_0} \leq K_c^{l_1}$, et
- $min_{c,l_1} \leq K_c^{l_1}$

Exemple 4.9 Ajoutons aux hypothèses de compositions d'interactions données dans l'Exemple 4.4 les contraintes en terme de \mathcal{P} relatives à leur observabilité soit $o(\neg i_x^1)$, $o(i_y^1)$ et $o(i_y^2)$.

Pour la composition d'interaction i_y^1 deux couples de conditions $(Cellc_{y,S}^{l_0}, Cellc_{y,S}^{l_1})$ respectent les conditions d'observabilité. L'un d'eux est $l_0 = 1$ et $l_1 = 3$ avec $min_{y,l_0} = 0$, $max_{y,l_0} = 1$, $min_{y,l_1} = 0$, $max_{y,l_1} = 1$, et l'autre est $l_0 = 2$ et $l_1 = 4$ avec $min_{y,l_0} = 2$, $max_{y,l_0} = 2$, $min_{y,l_1} = 2$, $max_{y,l_1} = 2$. Ainsi, l'observabilité de i_y^1 est équivalente à $(K_y^1 < K_y^3 \wedge K_y^1 \leq 1 \wedge 0 \leq K_y^3) \vee (K_y^2 < K_y^4 \wedge K_y^2 \leq 2 \wedge 2 \leq K_y^4)$, qui est équivalent à $(K_y^1 < K_y^3) \vee (K_y^2 < K_y^4 \wedge K_y^4 = 2)$ étant donné que les paramètres K_y^l sont tous compris dans l'intervalle $[0, 2]$.

Pour la composition d'interaction i_y^2 deux couples de conditions $(Cellc_{y,S}^{l_0}, Cellc_{y,S}^{l_1})$ respectent les conditions de l'observabilité. L'un d'eux est $l_0 = 1$ et $l_1 = 2$ avec $min_{c,l_0} = 0$, $max_{c,l_0} = 1$, $min_{c,l_1} = 2$, $max_{c,l_1} = 2$, et l'autre est $l_0 = 3$ et $l_1 = 4$ avec $min_{c,l_0} = 0$, $max_{c,l_0} = 1$, $min_{c,l_1} = 2$, $max_{c,l_1} = 2$. Ainsi, l'observabilité de i_y^2 est équivalente à $(K_y^1 < K_y^2 \wedge K_y^1 \leq 1 \wedge 2 \leq K_y^2) \vee (K_y^3 < K_y^4 \wedge K_y^3 \leq 1 \wedge 2 \leq K_y^4)$, qui est équivalent à $(K_y^1 < K_y^2 \wedge K_y^2 = 2) \vee (K_y^3 < K_y^4 \wedge K_y^4 = 2)$.

Pour l'unique composition d'interaction $(\neg i_x^1)$ sur x la contrainte d'observabilité est $K_x^2 < K_x^1$.

Pour terminer cette section nous proposons l'Exemple 4.10 qui illustre l'effet de l'introduction de propriétés sur le réseau introduit dans la Table 3.1. De la même manière que pour l'additivité nous adoptons dans le langage $LComp$ un abus de langage pour l'ajout d'une propriété d'observabilité $o(I)$ ajoutée à un ensemble de compositions d'interactions E_c noté naturellement $E_c \wedge o(I)$. Cet abus de langage consiste à remplacer le couple (I, q) de E_c par $(o(I), q)$. De plus, la notation $ao(I)$ indique qu'une composition d'interactions I est additive et observable (cette notation peut aussi apparaître dans E_c par un abus de langage).

Exemple 4.10 Ajoutons les hypothèses de compositions d'interactions $\{(ao(\neg i_x^1), 1)\}$ pour x , et $\{(ao(i_y^1), 1), (ao(i_y^2), 2)\}$ pour y au modèle \mathcal{M} de la Table 3.1. Nommons \mathcal{M}' le modèle obtenu. Les contraintes associées à ces hypothèses sont toutes celles qui sont produites dans les Exemples

hypothèses	contraintes
$E_x = \{(ao(\dot{i}_x^1), 1)\}$	$K_x^2 \leq K_x^1$ $K_x^2 < K_x^1$
$E_y = \{(ao(i_y^1), 1), (ao(i_y^2), 2)\}$	$(K_y^1 \leq K_y^3) \wedge (K_y^2 \leq K_y^4)$ $(K_y^1 \leq K_y^2) \wedge (K_y^3 \leq K_y^4)$ $(K_y^1 < K_y^3) \vee (K_y^2 < K_y^4 \wedge K_y^4 = 2)$ $(K_y^1 < K_y^2 \wedge K_y^2 = 2) \vee (K_y^3 < K_y^4 \wedge K_y^4 = 2)$

TAB. 4.1 – Hypothèses et contraintes de composition d’interactions associées au modèle \mathcal{M} de la Table 3.1 définissant le modèle \mathcal{M}' .

4.7, 4.8 et 4.9. Ces contraintes sont redonnées dans la Table 4.1. Ces contraintes induisent au total 8 valuations de l’ensemble \mathcal{P} des paramètres cinétiques discrets et de l’ensemble \mathcal{T} de transitions.

La Figure 4.2 montre ces 8 valuations où \mathcal{P} est représenté par $\{K_x^1, K_x^2, K_y^1, K_y^2, K_y^3, K_y^4\}$. Dans certains cas plusieurs valuations de \mathcal{P} produisent la même valuation de \mathcal{T} . Nous choisissons de rassembler ces valuations équivalentes en terme de \mathcal{T} . Nous le faisons en adoptant le symbole “.” entre valeurs de paramètres K pour représenter une disjonction entre ces valeurs qui offrent la même dynamique \mathcal{T} .

Nous obtenons alors 6 modèles dont \mathcal{T} est instancié, notés $\mathcal{M}^i, i \in \{A, B, C, D, E, F\}$ égaux à \mathcal{M} complété d’hypothèses de valuation ou de restriction des domaines des paramètres K_c^l . Pour exemple, pour \mathcal{M}^A le paramètre cinétique discret K_y^2 est contraint à être dans l’intervalle $[0, 1]$, $K_y^4 = 2$ et \mathcal{T} possède entre autres les transitions $\langle 1, 0 \rangle \rightarrow \langle 1, 0 \rangle$ et $\langle 1, 1 \rangle \rightarrow \langle 1, 0 \rangle$.

Remarquons qu’une contrainte d’observabilité d’une composition d’interactions sur c n’est pas dépendante de l’ensemble complet des compositions d’interactions sur c . De la même manière que pour l’additivité il est alors possible d’ajouter une contrainte d’observabilité sans connaître l’ensemble complet de compositions d’interactions.

4.2 Propriétés sur les comportements dynamiques

En général, les requêtes que l’on souhaite exprimer n’ont pas pour objet d’obtenir ou d’imposer des propriétés sur la totalité de \mathcal{T} , mais plutôt sur certains *chemins* de \mathcal{T} . Un chemin est compris comme une successions d’états.

En Sous-Section 4.2.1, nous présentons d’une façon générale les contraintes relatives à l’existence d’un chemin. Les Sous-Sections 4.2.2, 4.2.3 et 4.2.4 présentent les contraintes relatives à l’existence de certains cas particuliers de comportements, respectivement un état stationnaire, un chemin d’atteignabilité et un cycle. Ces trois Sous-Sections illustrent l’utilisation sophistiquée des contraintes proposées en Sous-Section 4.2.1. Notons que les possibilités d’expression ne sont pas limitées à ces trois cas particuliers de type de comportements. L’utilisateur peut imposer toutes les contraintes qu’il souhaite sur les chemins dont il exprime l’existence. Finalement, la Sous-Section 4.2.5 compare le pouvoir d’expression du langage proposé qui est basé sur la logique du premier ordre avec celui des formules temporelles de la logique CTL.

Exemple 4.11 *Considérons le modèle \mathcal{M}^C (défini dans l'Exemple 3.8). La requête $path(\mathcal{M}^C, Path, \mathcal{P})$, $Path = [\langle 1, 2 \rangle, \langle 0, 2 \rangle, \langle 0, 1 \rangle, \langle 0, 1 \rangle, \langle 0, 1 \rangle, \langle 0, 1 \rangle]$ répond positivement signifiant que l'ensemble des contraintes posées est cohérent. Nous pouvons le vérifier en constatant l'existence de ce chemin dans le graphe de transition \mathcal{T} correspondant à \mathcal{M}^C de la Figure 4.2.*

Exemple 4.12 *Par contre la requête $path(\mathcal{M}^C, Path, \mathcal{P})$, $Path = [\langle 1, 2 \rangle, \langle 0, 2 \rangle, \langle 0, 1 \rangle, \langle 1, 1 \rangle]$ répond négativement, indiquant une incohérence. Comme nous l'avons vu cette incohérence provient (au moins) du fait que \mathcal{T} n'accepte pas la dernière transition $\langle 0, 1 \rangle \rightarrow \langle 1, 1 \rangle$ du chemin $Path$ (voir les Exemples 3.9, 3.11 et 3.13).*

4.2.2 Contraintes d'existence d'un état stationnaire

Pour imposer qu'il existe un état S stationnaire pour un modèle \mathcal{M} et un ensemble \mathcal{P} de paramètres il suffit d'imposer

$$path(\mathcal{M}, Path, \mathcal{P}) \wedge Path = [S, S]$$

cette requête étant vraie si $S \rightarrow S$ est une transition de \mathcal{T} pour le modèle \mathcal{M} et les paramètres de \mathcal{P} (donc si S est stationnaire).

Exemple 4.13 *Considérons le modèle \mathcal{M}' de l'Exemple 4.10. La requête*

$$path(\mathcal{M}', Path, \mathcal{P}) \wedge Path = [S, S]$$

répond positivement. Cependant les variables S et \mathcal{P} ne sont pas instanciées dans la solution obtenue, nous ne sommes pas alors certain qu'il existe réellement un état stationnaire. Il peut être intéressant alors de déterminer tous les états stationnaires possibles. Pour cela nous procédons par énumération de toutes les valuations de S acceptant au moins une solution dont toutes les variables du problème sont instanciées (la Section A.2 en Annexe discute de cet aspect opérationnel de la PLC). Trois solutions sont obtenues avec S égal à $\langle 0, 1 \rangle$, $\langle 0, 2 \rangle$ et $\langle 1, 0 \rangle$. Nous pouvons le vérifier sur la Figure 4.2 qui donne en extension la relation entre \mathcal{T} et \mathcal{P} pour le modèle \mathcal{M}' .

4.2.3 Contraintes d'existence d'un chemin d'atteignabilité

Pour imposer qu'un état $S2$ soit atteignable à partir d'un état $S1$ en moins de N transitions pour un modèle \mathcal{M} et un ensemble \mathcal{P} de paramètre il suffit d'imposer $path(\mathcal{M}, Path, \mathcal{P})$ où $Path$ est de taille $N + 1$, le premier état de $Path$ est $S1$, et $\exists j \in [1, N + 1] \mid Path_j = S2$.

Le cas particulier où l'état $S2$ est d'autre part imposé stationnaire permet d'ajouter la contrainte redondante $j = N + 1$. Cela conduit à ne pas introduire de variable intermédiaire j car N est une constante.

Exemple 4.14 *Considérons le modèle \mathcal{M}' de l'Exemple 4.10. La requête*

$$\begin{aligned} path(\mathcal{M}', Path1, \mathcal{P}) \wedge Path1 = [S2, S2] \quad \wedge \\ path(\mathcal{M}', Path2, \mathcal{P}) \wedge Path2 = [S1, _, _, _, S2] \quad \wedge \\ S1_y = 0 \wedge S2_y = 2 \end{aligned}$$

permet d'imposer l'existence qu'un état $S2$ est atteignable en moins de 4 transitions à partir d'un état $S1$ avec les propriétés supplémentaires suivantes : $S1_y = 0$, $S2_y = 2$ et $S2$ est stationnaire

(la notation “_” indique une variable non unifié – ces variables servent ici à fixer la taille du chemin $Path_2$). Cette requête répond positivement. Cependant toutes les variables ne sont pas instanciées dans la solution obtenue, nous ne sommes pas alors certain qu’il existe réellement un chemin $Path_2$ satisfaisant les contraintes. Il peut être intéressant alors de déterminer tous les chemins $Path_2$ possibles. Nous obtenons par énumération deux solutions avec $Path_2$ égal à :

$$\begin{aligned} & [\langle 0, 0 \rangle, \langle 1, 0 \rangle, \langle 1, 1 \rangle, \langle 1, 2 \rangle, \langle 0, 2 \rangle] \text{ et} \\ & [\langle 1, 0 \rangle, \langle 1, 1 \rangle, \langle 1, 2 \rangle, \langle 0, 2 \rangle, \langle 0, 2 \rangle] \end{aligned}$$

Nous pouvons le vérifier sur la Figure 4.2 en constatant que ces deux chemins sont acceptés pour les modèles \mathcal{M}^E et \mathcal{M}^F et qu’ils n’existent pas d’autres chemins répondant aux contraintes posées.

4.2.4 Contraintes d’existence d’un cycle

Pour imposer l’existence d’un cycle de longueur maximale N pour un modèle \mathcal{M} et un ensemble \mathcal{P} de paramètre il suffit d’imposer $path(\mathcal{M}, Path, \mathcal{P})$ où $Path$ est de taille $N + 1$ et respecte la contrainte : $\exists j \in [2, N + 1] \mid Path_1 = Path_j$.

Il existe beaucoup de cas particuliers à ce type de contraintes selon les caractéristiques que nous souhaitons imposer au cycle. Les deux suivants sont très courant :

- Si le cycle doit contenir un état S bien spécifié la contrainte $\exists j' \in [1, j] \mid Path_{j'} = S$ doit être rajoutée.
- Si le cycle doit être exactement de taille N il faut ajouter la contrainte $j = N + 1 \wedge \forall j' \in [2, N] \mid Path_1 \neq Path_{j'}$.

Exemple 4.15 *Considérons le modèle \mathcal{M}' de l’Exemple 4.10. La requête*

$$\begin{aligned} & path(\mathcal{M}', Path, \mathcal{P}) \wedge Path = [S1, S2, S3, S4, S5, S6, S7] \wedge \\ & (S1 \neq S2) \wedge \\ & (S1 = S3 \Leftrightarrow B3) \wedge \\ & (S1 = S4 \Leftrightarrow B4) \wedge \\ & (S1 = S5 \Leftrightarrow B5) \wedge \\ & (S1 = S6 \Leftrightarrow B6) \wedge \\ & (S1 = S7 \Leftrightarrow B7) \wedge \\ & (B3 \vee B4 \vee B5 \vee B6 \vee B7) \end{aligned}$$

permet d’imposer l’existence d’un cycle qui n’est pas un état stationnaire ($S1 \neq S2$) en considérant un chemin de longueur 7 étant donné que le plus grand cycle possible a priori est de taille 6 (le nombre d’états de l’espace de concentrations discrètes pour le modèle \mathcal{M}'). Cette requête répond positivement. Cependant toutes les variables ne sont pas instanciées dans la solution obtenue, nous ne sommes pas alors certain qu’il existe réellement un chemin $Path_2$ satisfaisant les contraintes. Il peut être intéressant alors de déterminer tous les chemins $Path$ possibles. Nous obtenons par énumération six solutions avec $Path$ égal à :

$$\begin{aligned} & [\langle 1, 0 \rangle, \langle 1, 1 \rangle, \langle 0, 1 \rangle, \langle 0, 0 \rangle, \langle 1, 0 \rangle, \langle 1, 1 \rangle, \langle 0, 1 \rangle], \\ & [\langle 1, 0 \rangle, \langle 1, 1 \rangle, \langle 0, 1 \rangle, \langle 0, 0 \rangle, \langle 1, 0 \rangle, \langle 1, 1 \rangle, \langle 1, 2 \rangle], \\ & [\langle 0, 1 \rangle, \langle 0, 0 \rangle, \langle 1, 0 \rangle, \langle 1, 1 \rangle, \langle 0, 1 \rangle, \langle 0, 0 \rangle, \langle 1, 0 \rangle], \\ & [\langle 0, 0 \rangle, \langle 1, 0 \rangle, \langle 1, 1 \rangle, \langle 0, 1 \rangle, \langle 0, 0 \rangle, \langle 1, 0 \rangle, \langle 1, 1 \rangle], \\ & [\langle 1, 1 \rangle, \langle 0, 1 \rangle, \langle 0, 0 \rangle, \langle 1, 0 \rangle, \langle 1, 1 \rangle, \langle 0, 1 \rangle, \langle 0, 0 \rangle] \text{ et} \\ & [\langle 1, 1 \rangle, \langle 0, 1 \rangle, \langle 0, 0 \rangle, \langle 1, 0 \rangle, \langle 1, 1 \rangle, \langle 1, 2 \rangle, \langle 0, 2 \rangle] \end{aligned}$$

correspondant tous au même cycle $\langle 0,0 \rangle \rightarrow \langle 1,0 \rangle \rightarrow \langle 1,1 \rangle \rightarrow \langle 0,1 \rangle \rightarrow \langle 0,0 \rangle$. Nous pouvons le vérifier sur la Figure 4.2 en constatant que ce cycle est le seul cycle non stationnaire accepté.

4.2.5 Comparaison du pouvoir d'expression de la logique du premier ordre et de la logique CTL dans notre contexte d'étude

La logique CTL (*Computational Tree Logic*) est couramment utilisée pour exprimer des propriétés sur les comportements dans le cadre de l'analyse de réseaux de régulation génétique [8, 6]. Cette logique est basée sur les modèles de Kripke qui étendent la logique propositionnelle en introduisant les notions de mondes et d'accessibilités entre les mondes. À chaque monde est associé l'ensemble des propositions atomiques vraies dans ce monde. La manière dont est utilisée la logique CTL consiste à définir un monde pour chaque état de concentration et créer une relation d'accessibilité entre ces mondes figurant l'ensemble des transitions réalisables.

Expression de CTL en contraintes

Il est à noter qu'il est facile d'exprimer en PLC des expressions CTL sans la modalité "A" (pour tout chemin). En effet la modalité "E" exprimant l'existence d'un chemin correspond directement à l'opérateur existentiel attaché à toute variable introduite dans une requête en PLC.

D'autre part une propriété d'existence d'un chemin discret particulier est équivalente à l'existence du chemin continu correspondant, car la discrétisation de Thomas conserve les dynamiques continues. Par contre une propriété portant sur tous les chemins discrets n'est pas équivalente systématiquement à la même propriété sur tous les chemins continus, car la discrétisation de Thomas considère potentiellement des chemins discrets qui n'ont pas de contrepartie continue¹³. Ainsi il est important de noter que l'utilisation de la modalité "A" est réservée à des requêtes théoriques sur le réseau discret, mais que nous perdons alors le lien avec la dynamique continue du réseau de régulation génétique.

Limite du pouvoir d'expression de CTL

La logique propositionnelle est utilisable au niveau de chaque monde dans la logique CTL. Par contre la logique propositionnelle n'est pas utilisable pour exprimer des propriétés entre les mondes (ce sont uniquement les opérateurs spécifique à CTL qui sont utilisable pour exprimer des propriétés entre mondes). Ainsi, il n'est pas possible d'introduire des variables communes à différents mondes. Dans le contexte d'utilisation de la logique CTL précédemment décrit il est donc impossible d'exprimer formellement et partiellement la relation d'accessibilité entre les mondes. Donc cette utilisation de la logique CTL impose de connaître les valeurs des paramètres de \mathcal{P} afin de pouvoir définir la relation d'accessibilité, c'est-à-dire l'ensemble \mathcal{T} des transitions dans notre contexte.

Par ailleurs, il devient aussi impossible d'exprimer des propriétés sur des chemins dont au moins un état S est défini par rapport aux autres états du chemin, à moins de passer par une énumération des valuations de l'état S .

Exemple 4.16 *Il n'est pas possible en CTL d'exprimer sans énumération l'existence d'un état stationnaire quelconque comme dans la requête de l'Exemple 4.13. En effet le chemin dont l'existence est imposée possède un premier état qui est définis par rapport au deuxième état. La formule*

¹³Ceci est dû au non déterminisme étant donné l'abstraction considérée.

équivalente en CTL serait la suivante :

$$\begin{aligned} & ((x = 0 \wedge y = 0) \Rightarrow (EX (x = 0 \wedge y = 0))) \wedge \\ & ((x = 0 \wedge y = 1) \Rightarrow (EX (x = 0 \wedge y = 1))) \wedge \\ & ((x = 0 \wedge y = 2) \Rightarrow (EX (x = 0 \wedge y = 2))) \wedge \\ & ((x = 1 \wedge y = 0) \Rightarrow (EX (x = 1 \wedge y = 0))) \wedge \\ & ((x = 1 \wedge y = 1) \Rightarrow (EX (x = 1 \wedge y = 1))) \wedge \\ & ((x = 1 \wedge y = 2) \Rightarrow (EX (x = 1 \wedge y = 2))) \end{aligned}$$

où x et y sont les concentrations discrètes des protéines du système.

Ces énumérations pour exprimer des propriétés comportementales en CTL sont handicapantes si l'espace des états est vaste, et si nous souhaitons exprimer des contraintes telles que l'existence de plusieurs états stationnaires différents, l'existence de cycles ou d'atteignabilités de grandes dimensions, etc...

Fonctionnalités et applications

Cette partie présente les fonctionnalités de l'outil GNBox et une illustration des possibilités de l'outil sur trois applications biologiques. La palette de fonctionnalités présentées dans le Chapitre 5 illustre la facilité d'extension de l'outil GNBox à partir de son noyau constitué de la formalisation de Thomas en contraintes présentée dans le Chapitre 3. Les fonctionnalités proposent une large paramétrisation de la construction et de la représentation d'un modèle \mathcal{M} pour exprimer un large choix de type de requêtes : cohérence de l'ensemble des hypothèses, identification de comportements spécifiques, prédiction de propriétés tel que les relâchements nécessaires de contraintes dans le cas d'une inconsistance et tout cela pour des hypothèses pouvant porter sur plusieurs modèles similaires qui diffèrent par des mutations de certains gènes, sur des connaissances partielles sur les ordres des seuils de discrétisation et (comme présenté dans la précédente partie) sur \mathcal{T} , \mathcal{P} , la manière de composer les interactions de \mathcal{G} et les comportements.

Après ce chapitre nous montrons la faisabilité de l'approche contraintes sur trois applications biologiques réelles dans le Chapitre 6.

Chapitre 5

Fonctionnalités de l’outil GNBox

Dans les deux chapitres précédents nous avons présenté le noyau de l’outil GNBox qui permet de tester la cohérence d’hypothèses portant sur la connectivité d’un réseau génétique, les compositions d’interactions, les comportements et sur les paramètres cinétiques discrets. Remarquons que l’outil permet donc aussi d’énumérer les solutions cohérentes, en terme de \mathcal{T} pour l’aspect simulation et en terme de \mathcal{P} pour l’aspect inférence. Nous illustrons dans ce qui suit la facilité d’extension de l’outil à partir de ce noyau. Nous exposons les extensions suivantes : proposition automatique de relâchements d’hypothèses pour rétablir la consistance d’un ensemble d’hypothèses incohérent et plus généralement la prédiction de propriétés (Section 5.1), définition de modèles de différents types “mutants” d’un même réseau et lien entre eux (Section 5.2), introduction de paramètres d’entrées indiquant des concentrations constantes de protéines (Section 5.3), introduction de paramètres de seuils et de fonctionnalités associées (Section 5.4), et recherche d’ensembles cohérents de compositions d’interactions additives et observables (Section 5.5).

Tous les exemples de ce chapitre sont basés sur l’Exemple 4.10 constitué du graphe \mathcal{G} et modèle associé \mathcal{M} de la Table 3.1 avec pour hypothèses de compositions d’interactions $\{(ao(\dot{-}i_x^1), 1)\}$ pour x et $\{(ao(i_y^1), 1), (ao(i_y^2), 2)\}$ pour y .

5.1 Relâchement de contraintes et prédictions complètes

Dans le cas où l’ensemble d’hypothèses considéré est incohérent il est souhaitable de déterminer les relâchements d’hypothèses possibles permettant de lever l’incohérence. La détermination de l’ensemble complet des relâchements d’hypothèses possibles offre alors un théorème à l’ensemble des hypothèses.

Plus précisément il s’agit de déterminer des hypothèses “relachables” parmi les hypothèses dont nous sommes le moins sûr. Nous demandons pour cela au modélisateur/biologiste de séparer l’ensemble d’hypothèses posées, noté C , en deux sous-ensembles C^1 et C^2 : C^1 contient les contraintes dans lesquelles le biologiste a le plus confiance (typiquement celles dérivées de données expérimentales fiables) et C^2 les hypothèses qu’il souhaite remettre en question. C^2 est formé des hypothèses C_i : $C^2 \Leftrightarrow \bigwedge_i C_i$.

A chaque hypothèse C_i un Booléen B_i est défini tel que $B_i \Leftrightarrow C_i$. Notons C^3 la contrainte associée à ces associations : $C^3 \Leftrightarrow \bigwedge_i (B_i \Leftrightarrow C_i)$. Nous avons donc : $C \Leftrightarrow (C^1 \wedge C^3 \wedge \bigwedge_i B_i)$. Finalement, déterminer un sous-ensemble de contraintes de C^2 à relâcher revient à déterminer une valuation des Booléens B_i telle que $C^1 \wedge C^3$ soit cohérent.

Dans l’optique de trouver des théorèmes à $C^1 \wedge C^3$, ce que nous nommons *prédictions com-*

plètes, nous devons déterminer des propriétés sur l'ensemble complet des valuations des Booléens B_i telles que $C^1 \wedge C^3$ soit cohérent. Trois caractéristiques nécessaires d'un tel ensemble complet de valuations des Booléens B_i nous semblent intéressantes :

- $R0$ qui est la disjonction de toutes les valuations des B_i satisfaisant $C^1 \wedge C^3$,
- $R1$ qui est la disjonction de toutes les valuations des B_i satisfaisant $C^1 \wedge C^3$ et réduites aux B_i négatifs,
- $R2$ qui est la disjonction de toutes les valuations des B_i satisfaisant $C^1 \wedge C^3$ et réduites aux B_i positifs.

Ainsi $(C^1 \wedge C^3) \Rightarrow R0 \Rightarrow (R1 \wedge R2)$.

L'intérêt principal des propriétés $R1$ et $R2$ est de proposer des interprétations aisées de $R0$, en terme d'hypothèses nécessairement fausses pour $R1$ et en terme d'hypothèses nécessairement vraies pour $R2$.

Exemple 5.1 *Considérons l'ensemble formé des hypothèses données dans l'Exemple 4.10 et de l'hypothèse supplémentaire affirmant l'existence de trois états stationnaires distincts dans \mathcal{T} . Cet ensemble C de contraintes est inconsistant. Nous pouvons le vérifier "à la main" grâce à la Figure 4.2 qui donne la relation $r^{\mathcal{M}}(\mathcal{T}, \mathcal{P})$ en extension et graphiquement pour les hypothèses de l'Exemple 4.10. Aucune valuation de \mathcal{T} possédant 3 états stationnaires n'apparaît dans cette figure.*

Supposons que C^2 , l'ensemble des hypothèses remises en question soit constitué des six hypothèses d'additivité et d'observabilité de compositions d'interactions (C^1 est alors le reste des hypothèses, $C \Leftrightarrow C^1 \wedge C^2$). La contrainte C^3 obtenue en introduisant un Booléen B_i à chacun des éléments de C^2 est la suivante :

$$C^3 \Leftrightarrow ((B_1 \Leftrightarrow a(\dot{\neg}i_x^1)) \wedge (B_2 \Leftrightarrow o(\dot{\neg}i_x^1)) \wedge \\ (B_3 \Leftrightarrow a(i_y^1)) \wedge (B_4 \Leftrightarrow o(i_y^1)) \wedge \\ (B_5 \Leftrightarrow a(i_y^2)) \wedge (B_6 \Leftrightarrow o(i_y^2)))$$

Seules les valuations suivantes des Booléens B_i permettent de rendre cohérent $C^1 \wedge C^3$:

$$[B_1, B_2, B_3, B_4, B_5, B_6] = [0, 0, 1, 1, 1, 1] \\ [B_1, B_2, B_3, B_4, B_5, B_6] = [1, 1, 0, 0, 1, 1]$$

Nous obtenons alors les propriétés nécessaires $R0$, $R1$ et $R2$ suivantes :

$$R0 \Leftrightarrow (\neg B_1 \wedge \neg B_2 \wedge B_3 \wedge B_4 \wedge B_5 \wedge B_6 \vee \\ B_1 \wedge B_2 \wedge \neg B_3 \wedge \neg B_4 \wedge B_5 \wedge B_6) \\ R1 \Leftrightarrow (\neg B_1 \wedge \neg B_2 \vee \\ \neg B_3 \wedge \neg B_4) \\ R2 \Leftrightarrow (B_3 \wedge B_4 \wedge B_5 \wedge B_6 \vee \\ B_1 \wedge B_2 \wedge B_5 \wedge B_6)$$

$R1$ propose une interprétation aisée en terme d'hypothèses nécessairement fausses : "l'additivité et l'observabilité de $(\dot{\neg}i_x^1)$ sont nécessairement fausses, ou bien l'additivité et l'observabilité de i_y^1 sont nécessairement fausses" (vis-à-vis des contraintes posées).

$R2$ propose une interprétation aisée en terme d'hypothèses nécessairement vraies : "l'additivité et l'observabilité de i_y^1 et i_y^2 sont nécessairement vraies, ou bien l'additivité et l'observabilité de $(\dot{\neg}i_x^1)$ et i_y^2 sont nécessairement vraies" (vis-à-vis des contraintes posées).

D'autre part $R1$ et $R2$ sont réduites au maximum en utilisant la propriété $A \vee (A \wedge B) \Leftrightarrow A$ (la condition $A \wedge B$ étant donc inutile) afin qu'elles proposent une interprétation la plus aisée possible. La mise en œuvre de l'obtention de ces prédictions complètes "minimales" est décrite en Section 7.4.

Exemple 5.2 *Considérons l'ensemble des hypothèses donné dans l'Exemple 4.10 et de l'hypothèse supplémentaire d'existence de deux états stationnaires $S1$ et $S2$ dans \mathcal{T} tel que $S1_x < S2_x$ et $S1_y < S2_y$. Cet ensemble de contraintes est inconsistant.*

Nous considérons le même sous-ensemble de contraintes à remettre en question (C^2) que dans l'Exemple 5.1, et nous conservons du même coup les associations des Booléens B_i aux contraintes d'additivités et d'observabilités. C^3 , B_1 , B_2 , B_3 , B_4 , B_5 et B_6 sont alors définis de la même manière que dans l'Exemple 5.1.

Parmi toutes les valuations des Booléens B_i , seules les valuations suivantes permettent d'obtenir une cohérence de ($C^1 \wedge C^3$) (les valuations de $[B_1, B_2, B_3, B_4, B_5, B_6]$ sont classées dans l'ordre décroissant de la somme Sum des B_i) :

Sum	6	5	4	3	2	1	0
			[0, 0, 1, 1, 1, 1]	[0, 0, 0, 1, 1, 1] [0, 0, 1, 0, 1, 1] [0, 0, 1, 1, 1, 0]	[0, 0, 0, 1, 0, 1] [0, 0, 1, 1, 0, 0]	[0, 0, 0, 1, 0, 0]	

Les propriétés nécessaires $R1$ et $R2$ obtenues automatiquement sont alors les suivantes :

$$\begin{aligned}
 R1 &\Leftrightarrow (\neg B_1 \wedge \neg B_2) \\
 R2 &\Leftrightarrow (B_4 \vee B_3 \wedge B_5 \wedge B_6)
 \end{aligned}$$

Il est intéressant de remarquer la propriété suivante de monotonie :

$$\forall C_{sup} (C \Rightarrow R0 \Rightarrow (R1 \wedge R2)) \Rightarrow ((C \wedge C_{sup}) \Rightarrow R0 \Rightarrow (R1 \wedge R2))$$

signifiant que l'ajout d'hypothèses (C_{sup}) permet de conserver les propriétés nécessaires inférées sur le même ensemble d'hypothèses C^2 remis en question.

Des illustrations d'obtention de prédictions complètes sont données dans les Sous-Sections 6.1.2, 6.2.2 et 6.3.2.

5.2 Définition de mutants

Si nous possédons des observations concernant certains mutants il est évidemment intéressant de les ajouter à l'ensemble des hypothèses. Il faut donc pouvoir faire intervenir différents mutants dans une même requête. Les modèles de mutants sont construits de manière identique au modèle sauvage (le modèle d'origine construit comme dans la Section 3.1) à trois différences près :

- sont introduites les contraintes spécifiant que les paramètres K_c^l d'un gène c muté à une valeur v sont tous égaux à v (si $v = 0$ la mutation correspond à un *knock-out* ou KO du gène c , si $v \neq 0$ la mutation correspond à une "expression ectopique" du gène c à la valeur v). Cela exprime la suppression dans le graphe \mathcal{G} d'interactions des modèles mutants toutes les arêtes dont les espèces mutées sont des cibles,
- sont supprimées les contraintes d'observabilité de toutes les compositions d'interactions concernant les espèces mutées (les paramètres K_c^l des gènes c mutés ayant tous la même valeur la Contrainte 4.3 ne peut être vérifiée),

- sont introduites les contraintes spécifiant que tous les paramètres K_c^l associés à des gènes c non mutés des modèles mutants sont égaux aux paramètres correspondants du modèle du sauvage.

Exemple 5.3 *Considérons l'ensemble des hypothèses donné dans l'Exemple 4.10 comme définissant un modèle de type sauvage, nommé \mathcal{M}' . Nous souhaitons définir un modèle pour le type mutant dont le gène y subit une expression ectopique de valeur 1 (la tendance de y étant alors constante égal à 1). Nous nommons ce modèle $\mathcal{M}^\#$. Notons K_c^l les paramètres cinétiques discrets pour le modèle du type sauvage et $K_c^{\#l}$ ceux pour le type mutant. La définition du modèle mutant est identique à celle du modèle sauvage à l'exception des faits suivants :*

- nous ajoutons les contraintes $\forall l \quad K_y^{\#l} = 1$,
- nous supprimons les contraintes d'observabilité des compositions d'interactions sur y pour le modèle du type mutant,
- nous ajoutons les contraintes $\forall l \quad K_x^l = K_x^{\#l}$.

Il est alors possible d'imposer (en plus des deux modèles) l'existence de comportements spécifiques pour le type sauvage et pour le type mutant dans une même requête, et ce pour des comportements qui peuvent même dépendre les uns des autres. Par exemple, "existe t'il un état stationnaire commun au deux modèles?". L'ensemble de contraintes résultant est consistant. Nous obtenons deux solutions en terme de l'ensemble \mathcal{P} de paramètre du modèle sauvage \mathcal{M}' , l'une correspondant au modèle instancié \mathcal{M}^C et l'autre au modèle instancié \mathcal{M}^F et ce pour le même état stationnaire commun $\langle 0, 1 \rangle$ (voir Figure 4.2).

5.3 Introduction dans \mathcal{P} de paramètres d'entrées

Dans certains cas il est souhaitable de faire intervenir dans les conditions $Cell_{c,S}^l$ du système d'équations focales d'un modèle une ou plusieurs espèces dont la concentration discrète est constante le long de tout chemin. Une telle espèce est nommée une *entrée*. Pour cela il est possible d'écrire une équation focale réalisant cette propriété :

$$F_{c,S} = \sum_{l=1}^{max_c-1} \begin{array}{l} K_c^0 \cdot \\ K_c^l \cdot \\ K_c^{max_c} \cdot \end{array} \quad \begin{array}{l} S_c < t_c^1 \\ S_c < t_c^{l+1} \\ S_c \geq t_c^{max_c} \end{array} \quad \wedge \quad \begin{array}{l} S_c \geq t_c^l \\ S_c \geq t_c^{max_c} \end{array} \quad +$$

avec comme contraintes $K_c^l = l$ et $t_c^p = p$. En terme de graphe \mathcal{G} cela revient à dire que le nœud c n'est influencé que par lui même et ce par max_c interactions (une pour chaque seuil t_c^p).

Une autre manière de prendre en compte une espèce d'entrée c est de ne pas considérer d'équation focale pour c mais de considérer un paramètre dit *paramètre d'entrée* dans \mathcal{P} donnant la valeur constante de concentration discrète de c . L'intérêt de cette manière de prendre en compte les espèces d'entrée réside dans le fait qu'il n'est pas nécessaire de poser des contraintes associé à l'équation focale de c , mais aussi dans le fait que la représentation des chemins peut être simplifiée en supprimant la composante c des états. Par contre il est à noter que la concentration de c pour un modèle donné est alors constante quelque soit le chemin, et non plus seulement constante le long d'un chemin donné. Mais il est possible de définir autant de modèles que de valeurs souhaitées pour les concentrations des entrées en précisant bien que les autres paramètres (les K_c^l et paramètres de seuils que nous présentons en Section 5.4) de ces différents modèles sont égaux. Des illustrations d'utilisation de cette manière de prendre en compte les espèces d'entrées sont données dans les Sous-Sections 6.2.1 et 6.3.1.

5.4 Introduction dans \mathcal{P} de paramètres de seuils

Dans la démarche de modélisation discrète de réseaux génétiques il arrive de ne pas être certain de l'ordre entre les seuils et même du nombre de seuils nécessaire pour représenter les observations disponibles.

Dans la poursuite de notre démarche de généralisation de l'approche, nous proposons donc de prendre en compte formellement ces connaissances. Pour cela nous introduisons des *paramètres de seuils* t_c^p dans \mathcal{P} . Les paramètres t_c^p sont compris dans l'intervalle $[1, max_c]$ et pour tout $i \in [1, max_c]$ il existe un t_c^p égal i . Il est à noter que max_c est alors aussi une variable.

La formalisation est la suivante :

Contrainte 5.1

$$\bigwedge_c \left(\begin{array}{l} \bigwedge_p (1 \leq t_c^p \leq max_c) \wedge \\ \bigwedge_{i=0}^{max_c} \bigvee_p (t_c^p = i) \end{array} \right)$$

Nous illustrons dans l'Exemple 5.4 et en Sous-Section 6.3.2 l'utilité des variables t_c^p et max_c (pour toutes les autres requêtes de cette thèse ces variables sont connues).

Exemple 5.4 *Considérons l'ensemble des hypothèses donné dans l'Exemple 4.10 dont les contraintes $t_c^p = p$ sont retirées (ainsi les variables t_c^p et max_c ne sont plus instanciées systématiquement). Soit les hypothèses supplémentaires affirmant que l'état $\langle 1, 2 \rangle$ est stationnaire et que le nombre de seuils de l'espèce y est égal à 1 ($max_y = 1$). Il est à noter que pour l'espèce x il n'existe qu'un seuil, noté t_x^1 , et qu'il est donc égal à 1 de même que max_x étant donné la Contrainte 5.1.*

Cet ensemble est inconsistant, étant donné que l'état $\langle 1, 2 \rangle$ suppose au moins deux seuils distincts sur l'espèce y ($max_y \geq 2$) et que même avec deux seuils sur l'espèce y nous pouvons voir dans la Figure 4.2 que l'état $\langle 1, 2 \rangle$ ne peut être stationnaire d'après les hypothèses d'interactions.

Supposons que C^2 est l'ensemble de toutes les hypothèses d'additivité et d'observabilité des compositions d'interactions et de l'hypothèse "le nombre de seuils pour l'espèce y est strictement inférieur à 2". De la même manière que pour l'Exemple 5.1 nous ajoutons un Booléen B_i pour chacune des contraintes d'additivité et d'observabilité (B_1 à B_6) et un Booléen B_7 pour l'hypothèse sur le nombre de seuil de y . En utilisant le principe de relâchement (voir Section 5.1) nous obtenons la prédiction complète R1 suivante :

$$R1 \Leftrightarrow (\neg B_2 \wedge \neg B_7)$$

qui s'interprète par "l'observabilité de i_x^1 et l'hypothèse de réduction du nombre de seuils de y à 1 sont nécessairement fausses" (vis-à-vis des contraintes posées).

5.5 Recherche des ensembles cohérents de compositions d'interactions additives et observables

Une fonctionnalité intéressante dans l'optique *reverse-engineering* est l'inférence de propriétés sur les compositions d'interactions. Comme nous le précisons dans les Sections 4.1.2 et 4.1.3 il est très informatif de fournir à l'utilisateur des possibles compositions d'interactions qui sont à la fois additives et observables.

Nous proposons dans cette optique d'inférer les ensembles cohérents de compositions d'interactions additives et observables vis-à-vis d'un ensemble d'hypothèses. La mise en œuvre de ce processus d'inférence consiste d'abord à utiliser le langage *LComp* pour produire des ensembles de compositions d'interactions, puis à vérifier que ces ensembles de compositions d'interactions qui doivent être additives et observables sont cohérents.

Chapitre 6

Applications

Nous présentons dans ce chapitre trois applications pour construire et analyser des réseaux de régulation modélisés avec le formalisme de Thomas. L'analyse du réseau concernant le contrôle de l'immunité chez le bactériophage λ [50] (Section 6.1) illustre principalement l'expression de comportements d'atteignabilité et l'inférence de propriétés sur les compositions d'interactions. L'analyse du réseau concernant le stress nutritionnel carboné chez la bactérie *Escherichia coli* [42] (Section 6.2) illustre principalement le relâchement d'hypothèses pour l'obtention d'un modèle cohérent et l'obtention de prédictions complètes. L'analyse du réseau concernant le module *gap-gene* de la segmentation de l'embryon de la mouche *Drosophila melanogaster* [43] (Section 6.3) illustre principalement l'expression de comportement stationnaires pour différents types mutants dans différentes régions de l'embryon, et la recherche du nombre minimal de seuils nécessaire pour accepter toutes les hypothèses.

Ces applications sur des problèmes biologiques réels montrent l'intérêt du formalisme décrit dans les Chapitres 3 et 4 ainsi que celui de l'outil GNBox et des extensions décrites dans le Chapitre 5. Les performances de cet outil en réponse aux requêtes formulées dans ce Chapitre sont rassemblées dans la Section 8.4.

6.1 Application au contrôle de l'immunité par le phage λ

Le bactériophage λ (ou simplement phage λ) est un virus qui infecte la bactérie *Escherichia coli*. Cette infection consiste en l'injection du matériel génétique du virus dans le cytoplasme de la bactérie. Nous nous concentrons ici sur deux observations simples de l'évolution de la cellule bactérienne après infection par le phage λ :

- soit l'ADN viral s'introduit dans le matériel génétique de la bactérie et la cellule continue à se reproduire normalement,
- soit le matériel génétique viral se réplique pour créer de nouvelles particules virales jusqu'à la lyse (destruction) de la cellule, conduisant au largage des nouveaux virus dans le milieu extra-cellulaire.

Le premier cas correspond à la phase dite *lysogénique* tandis que le deuxième cas correspond à la phase dite *lytique*.

L'intérêt de ce problème biologique dans le contexte de notre travail est de montrer les capacités de GNBox à exprimer et imposer des comportements d'atteignabilité d'états et conséquemment à inférer des propriétés sur les paramètres \mathcal{P} permettant soit de compléter soit de remettre en question les hypothèses d'interaction. Le problème crucial de la taille des chemins

se pose ici et est développé en Sous-Section 6.1.1.

Nous présentons en Sous-Section 6.1.1 les hypothèses de modélisation et les observations biologiques tirées de [50] et exprimées sous forme de contraintes et en Sous-Section 6.1.2 les résultats concernant d'une part la fonctionnalité de base de notre outil, c'est-à-dire la cohérence de l'ensemble des hypothèses, et d'autre part l'inférence de prédictions complètes.

6.1.1 Le modèle et les hypothèses

La décision entre la phase lysogénique ou la phase lytique est établie par un contrôle subtil des gènes viraux. Dans le modèle proposé en [50], nous trouvons quatre de ces gènes viraux notés cI , cro , cII et n . Le gène cI est exprimé dans la phase lysogénique, tandis que c'est le gène cro qui est exprimé dans la phase lytique. Le graphe \mathcal{G} d'interactions entre ces gènes est donné par la Figure 6.1. Les interactions et les hypothèses de compositions de ces interactions sont données dans la Table 6.1 (colonnes 2 et 3 respectivement). Les équations focales résultantes des interactions supposées sont données en Annexe E Section E.1.

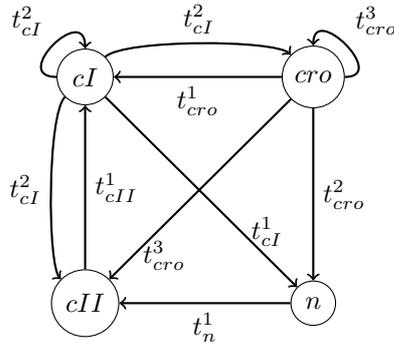


FIG. 6.1 – Graphe \mathcal{G} d'interactions pour le modèle du contrôle de l'immunité chez le phage λ .

espèce	interactions	hypothèses de compositions d'interactions
cI	$i_{cI}^1 = (cI, t_{cI}^2, cI)$ $i_{cI}^2 = (cro, t_{cro}^1, cI)$ $i_{cI}^3 = (cII, t_{cII}^1, cI)$	$\{ (ao(i_{cI}^1), 1),$ $(ao(\dot{-}i_{cI}^2), 2),$ $(ao(i_{cI}^3), 3) \}$
cro	$i_{cro}^1 = (cI, t_{cI}^2, cro)$ $i_{cro}^2 = (cro, t_{cro}^3, cro)$	$\{ (ao(\dot{-}i_{cro}^1), 1),$ $(ao(\dot{-}i_{cro}^2), 2) \}$
cII	$i_{cII}^1 = (cI, t_{cI}^2, cII)$ $i_{cII}^2 = (cro, t_{cro}^3, cII)$ $i_{cII}^3 = (n, t_n^1, cII)$	$\{ (ao(\dot{-}i_{cII}^1), 1),$ $(ao(\dot{-}i_{cII}^2), 2),$ $(ao(i_{cII}^3), 3) \}$
n	$i_n^1 = (cI, t_{cI}^1, n)$ $i_n^2 = (cro, t_{cro}^2, n)$	$\{ (ao(\dot{-}i_n^1), 1),$ $(ao(\dot{-}i_n^2), 2) \}$

TAB. 6.1 – Interactions et hypothèses de composition de ces interactions pour le modèle du contrôle de l'immunité chez le phage λ .

Un état S est représenté par un vecteur $\langle S_{cI}, S_{cro}, S_{cII}, S_n \rangle$ des concentrations discrètes des

protéines exprimées par les gènes viraux. Tous les seuils t_c^p sont supposés égaux à p ; ils sont donc ordonnés. Ainsi d'après \mathcal{G} , nous avons $max_{cI} = 2$, $max_{cro} = 3$, $max_{cII} = 1$ et $max_n = 1$. Il en résulte un espace d'états de concentration de $(2 + 1) * (3 + 1) * (1 + 1) * (1 + 1) = 48$ états.

La cellule non infectée ne possède aucune protéine virale et peut donc être représentée par l'état $S0 = \langle S0_{cI}, S0_{cro}, S0_{cII}, S0_n \rangle = \langle 0, 0, 0, 0 \rangle$.

Seul le gène cI est exprimé dans la phase lysogénique. Cette phase est formalisée par l'état stationnaire $S1 = \langle S1_{cI}, S1_{cro}, S1_{cII}, S1_n \rangle = \langle 2, 0, 0, 0 \rangle$ tel que la concentration de cI reste égale à sa plus haute valeur.

Seul le gène cro est exprimé dans la phase lytique. Cette phase est formalisée par un cycle de deux états $S2 = \langle S2_{cI}, S2_{cro}, S2_{cII}, S2_n \rangle = \langle 0, 2, 0, 0 \rangle$ et $S3 = \langle S3_{cI}, S3_{cro}, S3_{cII}, S3_n \rangle = \langle 0, 3, 0, 0 \rangle$, tels que la concentration de la protéine cro reste autour de ces plus hautes valeurs 2 et 3.

Ces observations sont formalisées selon [50].

En accord avec [50], l'observation de ces deux atteignabilités est formalisée par les contraintes suivantes :

$$\begin{aligned} & path(\mathcal{M}, [S1, S1], \mathcal{P}) \wedge \\ & path(\mathcal{M}, [S2, S3, S2], \mathcal{P}) \wedge \\ & path(\mathcal{M}, [S0, \dots, S1], \mathcal{P}) \wedge \\ & path(\mathcal{M}, [S0, \dots, S23], \mathcal{P}) \wedge \\ & S0 = \langle 0, 0, 0, 0 \rangle \wedge \\ & S1 = \langle 2, 0, 0, 0 \rangle \wedge \\ & S23 = \langle 0, S23_{cro}, 0, 0 \rangle \wedge \\ & S23_{cro} \in [2, 3] \end{aligned}$$

où le troisième et quatrième chemins pour l'atteignabilité des deux phases sont de taille 48 états (48 étant le nombre d'états de l'espace des états).

Une question intéressante est de connaître la taille L du plus long chemin sans cycle, appelé *diamètre du réseau*, associé à un ensemble d'hypothèses. Pour trouver ce diamètre il nous faut trouver l'entier L tel qu'il existe un chemin sans cycle de taille L et qu'il n'existe pas de chemin sans cycle de taille $L + 1$. Les hypothèses d'existence d'un chemin d'une longueur quelconque ayant une propriété ne dépendant pas de la taille peuvent être remplacés dans les requêtes par l'existence d'un chemin d'une longueur l ayant la même propriété dans chaque requête avec l la taille du chemin sans cycle. Ce diamètre permet alors de limiter la taille des chemins à considérer dans les requêtes. Pour ce réseau le diamètre est de 43 en tenant compte de toutes les hypothèses précédentes. Ce problème hautement combinatoire obtient une réponse en deux requêtes : l'une prouvant l'existence d'une solution pour une taille de 43 en 135 secondes, l'autre prouvant l'inconsistance pour une taille de 44 en 977 secondes.

6.1.2 Cohérence et inférence de propriétés

L'outil GNBox permet de prouver la cohérence de l'ensemble de contraintes précédent en 6 secondes¹⁴. Et il permet d'obtenir les valuations de l'ensemble \mathcal{P} qui sont solutions. Cette fonc-

¹⁴Ces performances sont obtenues avec un ordinateur portable ayant 1 Go de RAM et tournant à 1.8 GHz. Nous choisissons d'arrondir les performances à la seconde près. Il est à noter de plus que seule 1 seconde parmi le temps total est nécessaire au solveur minisat pour trouver une solution à la formule SAT créée, le reste du temps est consacré à la création des contraintes CLP, de la formule Booléenne équivalente en CNF et le transfert de la

tionnalité basique est vraiment un grand avantage par aux autres formalismes/outils cherchant à instancier, “à la main”, les valeurs de paramètres respectant toutes leurs hypothèses.

Exemple 6.1 Une des valuations de \mathcal{P} obtenue qui est solution de l'ensemble des hypothèses est¹⁵ :

$$\begin{aligned} K_{cI}^1 &= 1, & K_{cI}^2 &= 1, & K_{cI}^3 &= 0, & K_{cI}^4 &= 1, & K_{cI}^5 &= 2, & K_{cI}^6 &= 2, & K_{cI}^7 &= 0, & K_{cI}^8 &= 1, \\ K_{cro}^1 &= 3, & K_{cro}^2 &= 1, & K_{cro}^3 &= 0, & K_{cro}^4 &= 0, & & & & & & & & \\ K_{cII}^1 &= 0, & K_{cII}^2 &= 1, & K_{cII}^3 &= 0, & K_{cII}^4 &= 0, & K_{cII}^5 &= 0, & K_{cII}^6 &= 0, & K_{cII}^7 &= 0, & K_{cII}^8 &= 0, \\ K_n^1 &= 1, & K_n^2 &= 0, & K_n^3 &= 0, & K_n^4 &= 0, & & & & & & & & \end{aligned}$$

De plus l'outil peut exhiber les ensembles de compositions d'interactions additives et observables cohérents avec un ensemble d'hypothèses (voir Section 5.5). L'Exemple 6.2 illustre cette fonctionnalité sur un cas simple où tous les paramètres sont connus et où nous nous limitons à trouver les ensembles minimaux de compositions d'interactions.

Exemple 6.2 Pour la valuation donnée dans l'Exemple 6.1 nous trouvons avec GNBox en 1 seconde un unique ensemble minimal de compositions d'interactions additives et observables pour l'espèce cII (car la valuation des paramètres K_{cII}^l montre que deux valeurs 0 et 1 et il n'y a donc besoin, au minimum, que d'une composition d'interactions pour expliquer ce découpage en deux valeurs) :

$\{ao(((\dot{\neg}i_{cII}^1) \wedge (\dot{\neg}i_{cII}^2)) \wedge i_{cII}^3))\}$. Pour l'espèce n nous trouvons un unique ensemble d'une seule composition d'interactions additive et observable : $\{ao(((\dot{\neg}i_n^1) \wedge (\dot{\neg}i_n^2)))\}$. Cependant nous trouvons 14 ensembles possibles de deux compositions d'interactions pour l'espèce cI et les 5 ensembles possibles de deux compositions d'interactions pour cro suivants :

$$\begin{aligned} & \{ ao((\dot{\neg}i_{cro}^1)), & & ao((\dot{\neg}i_{cro}^2)) \}, \\ & \{ ao((\dot{\neg}i_{cro}^1)), & & ao((i_{cro}^1 \dot{\vee} (\dot{\neg}i_{cro}^2))) \}, \\ & \{ ao((\dot{\neg}i_{cro}^1)), & & ao((\dot{\neg}i_{cro}^1) \wedge (\dot{\neg}i_{cro}^2)) \}, \\ & \{ ao((\dot{\neg}i_{cro}^1)), & & ao(((i_{cro}^1 \wedge i_{cro}^2) \dot{\vee} ((\dot{\neg}i_{cro}^1) \wedge (\dot{\neg}i_{cro}^2)))) \} \\ & \{ ao(((\dot{\neg}i_{cro}^1) \wedge i_{cro}^2)), & & ao((\dot{\neg}i_{cro}^1) \wedge (\dot{\neg}i_{cro}^2)) \}. \end{aligned}$$

Parmi l'ensemble de toutes les hypothèses d'additivité et d'observabilité de compositions d'interactions supposées C^2 il est possible avec notre outil de déterminer lesquelles sont nécessaires pour obtenir une consistance de toutes les autres contraintes C^1 (modèle, interactions, compositions d'interactions et comportements). Cette fonctionnalité est présentée dans la Section 5.1 comme la construction de la prédiction complète $R2$. Dans ce cas GNBox fournit en 5 secondes que seules deux contraintes de ces contraintes sont nécessaires pour expliquer toutes les hypothèses C^1 : $R2 \Leftrightarrow o((\dot{\neg}i_{cro}^1)) \wedge o((\dot{\neg}i_{cro}^2))$. Les observabilités des compositions d'interactions $(\dot{\neg}i_{cro}^1)$ et $(\dot{\neg}i_{cro}^2)$ sont donc nécessaires. Cela revient à faire du *reverse-engineering* de \mathcal{G} en tenant compte des restrictions faite sur le graphe d'interactions \mathcal{G} initial. Dans ce cas les deux flèches de \mathcal{G} arrivant sur le nœud cro sont absolument nécessaires pour expliquer le reste des hypothèses (celles de C^1 qui ne sont pas remise en question).

solution SAT sur les variables entières CLP permettant d'obtenir un résultat compréhensible.

¹⁵Rappelons que les indices l des paramètres cinétiques discrets K_c^l sont fixés à leur création (voir Section 3.1) vis-à-vis des indices r des interactions i_c^r . Les équations focales obtenues sont données en Section E.1.

6.2 Application au stress nutritionnel carboné chez la bactérie *Escherichia coli*

Escherichia coli est une bactérie dont la population augmente exponentiellement dans un milieu où il y a assez de carbone (sa nourriture). Dans le cas d'un stress carboné il n'y a plus assez de carbone et la population arrête d'augmenter pour survivre en conservant les quelques réserves de carbone encore présentes.

L'intérêt de ce problème biologique dans le contexte de notre travail réside dans la remise en question d'une modélisation infructueuse présentée en [42] et la proposition de modifications des hypothèses.

Nous présentons en Sous-Section 6.2.1 les hypothèses de modélisation et les observations biologiques tirées de [42], adaptées à notre formalisme et exprimées sous forme de contraintes. Nous présentons en Sous-Section 6.2.2 les résultats concernant le relâchement d'hypothèses d'additivité et d'observabilité, et ceux concernant l'inférence de prédictions complètes permettant de guider le biologiste dans ses choix d'hypothèses.

6.2.1 Le modèle et les hypothèses

Le stress est formalisé par une entrée us (voir Sous-Section 5.3) prenant une valeur Booléenne : 0 en l'absence du stress et 1 en présence du stress.

Le modèle utilisé possède 6 gènes : celui produisant la protéine C-réactive noté crp , celui produisant l'enzyme adénylate cyclase cya noté cya , fis noté fis , celui produisant la gyrase $gyrAB$ noté gyr , et celui produisant la topoisomérase $topA$ noté top .

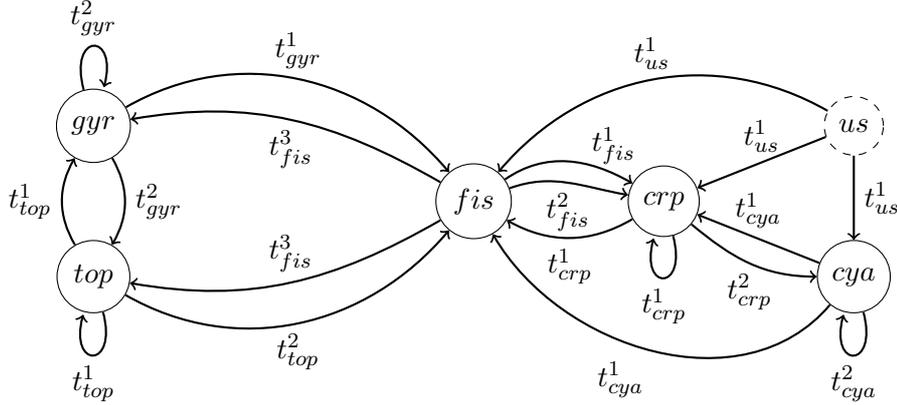
Le graphe \mathcal{G} d'interactions entre ces gènes est donné par la Figure 6.2 où l'entrée us est représentée par un cercle en pointillés. Les interactions et les hypothèses de compositions de ces interactions sont données dans la Table 6.2. Les équations focales résultantes des interactions supposées sont données en Annexe E Section E.2.

Il est à noter que nous n'imposons pas d'hypothèse d'observabilité pour l'unique composition d'interaction sur cya . En fait le modèle n'introduit pas le complexe cAMP/CRP afin d'obtenir un réseau purement génétique (non métabolique). Cela entraîne une autorégulation sur cya alors qu'en fait cya régule cAMP/CRP et c'est ce dernier qui régule cya . Cette régulation en "deux étapes" modélisée par une interaction en "une étape" entraîne la nécessité de ne pas considérer l'observabilité de l'auto-interaction sur cya .

Ce modèle est directement inspiré du papier [42].

Comme précédemment les seuils t_c^p sont égaux à p . Ainsi nous avons $max_{crp} = 2$, $max_{cya} = 2$, $max_{fis} = 3$, $max_{gyr} = 2$ et $max_{top} = 2$ (pour l'entrée us nous avons $max_{us} = 1$). Il en résulte un espace d'états de $(2 + 1) * (2 + 1) * (3 + 1) * (2 + 1) * (2 + 1) = 324$ états.

Si le stress est absent ($us = 0$) la cellule se dirige vers un état stationnaire, noté S^{ns} (ns pour "non stressé"). Si le stress est présent ($us = 1$) la cellule se dirige vers un état stationnaire, noté S^s (s pour "stressé").


 FIG. 6.2 – Graphe \mathcal{G} d'interactions pour le modèle du stress carboné chez *E. coli*.

De plus, suivant [42], nous ajoutons les hypothèses suivantes à propos de ces états :

$$\begin{aligned} S_{crp}^{ns} &= 1 & S_{crp}^s &= 2 \\ S_{cya}^{ns} &= 1 & S_{cya}^s &= 1 \\ S_{fis}^{ns} &= 3 & S_{fis}^s &= 0 \\ S_{gyr}^{ns} - S_{gyr}^s &> S_{top}^{ns} - S_{top}^s \end{aligned}$$

Il est à noter que les états S^{ns} et S^s ne sont pas complètement instanciés et qu'une relation les lie.

Deux comportements sont observés :

1. si l'état non stressé S^{ns} est perturbé par la présence de stress alors la cellule se dirige vers l'état stationnaire stressé S^s ,
2. si l'état stressé S^s est perturbé par l'absence de stress alors la cellule se dirige vers l'état stationnaire non stressé S^{ns} .

Nous définissons, comme présenté dans la Section 5.3, deux modèles M_{ns} et M_s , respectivement pour le modèle sans stress ($us = 0$) et pour le modèle avec stress ($us = 1$). Ils partagent donc les mêmes paramètres cinétiques discrets K_c^l . Leur seule différence d'un point de vue paramètres réside dans la valeur du paramètre d'entrée us .

Les deux comportements précédemment cités sont formalisés par les contraintes suivantes :

$$\begin{aligned} &path(M_{ns}, [S^{ns}, S^{ns}], \mathcal{P}) \wedge \\ &path(M_s, [S^s, S^s], \mathcal{P}) \wedge \\ &path(M_{ns}, [S^s, \dots, S^{ns}], \mathcal{P}) \wedge \\ &path(M_s, [S^{ns}, \dots, S^s], \mathcal{P}) \end{aligned}$$

où le troisième et quatrième chemins pour l'atteignabilité des deux états stationnaires sont de taille 324 états (le nombre d'états possibles du modèle). Nous faisons varier la taille de ces chemins dans les requêtes de la Sous-section 6.2.2 afin d'être capable de poser les contraintes.

6.2.2 Relâchement d'hypothèses d'additivité et d'observabilité de compositions d'interactions et inférence de propriétés

Grâce à GNBox nous prouvons que l'ensemble de contraintes précédent est inconsistant en 6 secondes pour une taille de chemin de 10 états, et 176 secondes pour une taille de chemin de

espèces	interactions	hypothèses de compositions d'interactions
<i>crp</i>	$i_{crp}^1 = (us, t_{us}^1, crp)$ $i_{crp}^2 = (crp, t_{crp}^1, crp)$ $i_{crp}^3 = (cya, t_{cya}^1, crp)$ $i_{crp}^4 = (fis, t_{fis}^1, crp)$ $i_{crp}^5 = (fis, t_{fis}^2, crp)$	$\{ (ao((i_{crp}^1 \wedge i_{crp}^2) \wedge i_{crp}^3)), 1),$ $(ao((\neg i_{crp}^4)), 2),$ $(ao((\neg i_{crp}^5)), 3) \}$
<i>cya</i>	$i_{cya}^1 = (us, t_{us}^1, cya)$ $i_{cya}^2 = (crp, t_{crp}^2, cya)$ $i_{cya}^3 = (cya, t_{cya}^2, cya)$	$\{ (a(((\neg i_{cya}^1) \vee (\neg i_{cya}^2)) \vee (\neg i_{cya}^3))), 1) \}$
<i>fis</i>	$i_{fis}^1 = (us, t_{us}^1, fis)$ $i_{fis}^2 = (crp, t_{crp}^1, fis)$ $i_{fis}^3 = (cya, t_{cya}^1, fis)$ $i_{fis}^4 = (gyr, t_{gyr}^1, fis)$ $i_{fis}^5 = (top, t_{top}^2, fis)$	$\{ (ao(((\neg i_{fis}^1) \vee (\neg i_{fis}^2)) \vee (\neg i_{fis}^3))), 1),$ $(ao((i_{fis}^4 \wedge (\neg i_{fis}^5))), 2) \}$
<i>gyr</i>	$i_{gyr}^1 = (fis, t_{fis}^3, gyr)$ $i_{gyr}^2 = (gyr, t_{gyr}^2, gyr)$ $i_{gyr}^3 = (top, t_{top}^1, gyr)$	$\{ (ao((\neg i_{gyr}^1)), 1),$ $(ao(((\neg i_{gyr}^2) \vee i_{gyr}^3)), 2) \}$
<i>top</i>	$i_{top}^1 = (fis, t_{fis}^3, top)$ $i_{top}^2 = (gyr, t_{gyr}^2, top)$ $i_{top}^3 = (top, t_{top}^1, top)$	$\{ (ao(i_{top}^1), 1),$ $(ao((i_{top}^2 \wedge (\neg i_{top}^3))), 2) \}$

 TAB. 6.2 – Interactions et hypothèses sur leurs compositions pour le modèle du stress carboné chez *E. coli*.

100 états. En fait juste l'imposition des deux états stationnaires donne une incohérence en moins d'une seconde, prouvant ainsi que l'ensemble de contraintes initial (avec les atteignabilités de taille 324) est inconsistant. En [42], il est mentionné que le modèle instancié proposé est bien inconsistant. Mais cela ne prouve pas qu'il n'existe pas de modèle instancié consistant avec les observations.

En vue de relâcher des contraintes pour obtenir une cohérence (voir Section 5.1) nous proposons de trouver les hypothèses nécessairement fausses parmi celles d'additivité et d'observabilité des compositions d'interactions (données dans la troisième colonne de la Table 6.2). Nous contruisons pour cela la prédiction complète $R1$ en introduisant des Booléens B_i pour chaque hypothèse C_i d'additivité et d'observabilité des compositions d'interactions.

Nous trouvons :

$$R1 \Leftrightarrow (\quad \neg a((\neg i_{gyr}^1)) \wedge \quad \neg a(((\neg i_{gyr}^2) \vee i_{gyr}^3)) \quad \vee \\
 \quad \neg a((\neg i_{gyr}^1)) \wedge \quad \neg o((\neg i_{gyr}^1)) \quad \vee \\
 \quad \neg a(i_{top}^1) \wedge \quad \neg a((i_{top}^2 \wedge (\neg i_{top}^3))) \quad \vee \\
 \quad \neg a(i_{top}^1) \wedge \quad \neg o(i_{top}^1) \quad)$$

en 55 secondes avec une taille de chemin de 10, et en 1380 secondes avec une taille de chemin de 100.

Après discussion avec le biologiste il semble non acceptable de relâcher l'additivité de la première composition d'interactions sur *gyr*, $(\neg i_{gyr}^1)$. Il nous faut alors construire la prédiction complète $R1$ en introduisant des Booléens B_i pour chaque hypothèse C_i d'additivité et d'obser-

vabilité sauf pour l'additivité de la première composition sur *gyr*. Il en résulte :

$$R1 \Leftrightarrow \left(\begin{array}{l} \neg a(i_{top}^1) \wedge \neg a((i_{top}^2 \wedge (\dot{-}i_{top}^3))) \vee \\ \neg a(i_{top}^1) \wedge \neg o(i_{top}^1) \end{array} \right)$$

Ceci suggère que la première composition d'interactions sur *top* (i_{top}^1) n'est pas additive et qu'il est alors possible d'observer une inhibition de *top* par *fis*. Il se trouve justement que cette inhibition est observée pour un autre stress [55]¹⁶ ce qui conforte donc cette nouvelle hypothèse pour le stress nutritionnel carboné.

Il semble finalement que les hypothèses de composition d'interactions sur *top* ne soient pas du tout justifiées. Dans cette optique nous proposons de déterminer les compositions d'interactions nécessairement observables en se limitant aux compositions d'interactions du type i_c^r et du type $(\dot{-}i_c^r)$. L'idée sous-jacente de cette limitation est de fournir un résultat facilement interprétable en terme du graphe \mathcal{G} d'interactions complété de signes sur les interactions (comme précédemment discuté en Sous-Section 1.2.1). D'autre part cette limitation permet tout de même de guider l'utilisateur sur le choix des hypothèses de compositions.

Pour cela nous ajoutons les hypothèses de compositions d'interactions suivantes sur *top* (celles sur les autres espèces restant identiques à la Table 6.2) :

$$\{(o(i_{top}^1), 1), (o(i_{top}^2), 2), (o(i_{top}^3), 3), (o((\dot{-}i_{top}^1)), 4), (o((\dot{-}i_{top}^2)), 5), (o((\dot{-}i_{top}^3)), 6)\}$$

Nous remettons en question ces six hypothèses d'observabilité de composition d'interactions constituant donc l'ensemble C^2 présenté dans la Section 5.1 et nous obtenons la formule nécessaire *R2* suivante :

$$R2 \Leftrightarrow \left(\begin{array}{l} o(i_{top}^2) \vee \\ o((\dot{-}i_{top}^1)) \end{array} \right)$$

14 secondes avec une taille de chemin de 10, et 362 secondes avec une taille de chemin de 100.

Cela indique que toute solution de l'ensemble de contraintes posé (excepté les hypothèses de compositions d'interactions sur *top*) possède soit la propriété $o(i_{top}^2)$, soit la propriété $o((\dot{-}i_{top}^1))$. Ce résultat peut constituer une information essentielle pour aider le biologiste à faire de nouvelles hypothèses sur les compositions d'interactions sur *top*.

6.3 Application au module *gap-gene* de la segmentation de l'embryon de *Drosophila melanogaster*

L'embryon de *Drosophila melanogaster* (ou simplement drosophile) se segmente en différentes régions composées de cellules dans des états spécifiques de concentrations de certaines protéines. La segmentation se déroule en plusieurs étapes successives et nous nous intéressons ici à la segmentation selon l'axe antérieur-postérieur de l'embryon géré par les gènes *gap*. Ces gènes forment le module de régulation génétique nommé *gap-gene*. Les travaux dont nous nous inspirons sont ceux décrits en [43] où est décrit un état stationnaire observé pour chaque région et ceci pour plusieurs mutants.

¹⁶Cette publication contient la citation suivante "when *Fis* levels are low, hydrogen peroxide treatment results in *topA* activation".

L'intérêt de ce problème biologique est double dans le contexte de notre travail. D'une part d'un point de vue expression nous avons à imposer des comportements stationnaires pour plusieurs mutants et plusieurs régions, et faire le lien entre tous les modèles associés. D'autre part la modélisation tirée de [43] peut être explorée vis à vis de cette grande quantité d'observations et ce en particulier pour répondre à un nouveau type de questions. Elles concernent le *nombre minimal de seuils nécessaires* pour expliquer des phénomènes biologiques dans le cadre de la formalisation sous la forme de réseaux de Thomas. Cette question a été posée par un des auteurs de [43] exprimant son incertitude.

Nous présentons en Sous-Section 6.3.1 les hypothèses de modélisation et les observations biologiques tirées de [43] et exprimées sous forme de contraintes. Nous présentons en Sous-Section 6.3.2 des résultats concernant le nombre minimal de seuils nécessaires pour satisfaire toutes les hypothèses.

6.3.1 Le modèle et les hypothèses

Le modèle du module *gap-gene* possède 7 gènes : giant noté *gt*, hunchback zygotique noté *hb_z*, kruppel noté *kr*, knirps noté *kni*, bicoid noté *bcd*, hunchback maternel noté *hb_m*, et caudal noté *cad*. Les gènes *bcd*, *hb_m* et *cad* sont des gènes d'entrée dont la concentration ne varie pas et est déterminée par l'organisme mère. Ainsi, la concentration constante de chacun de ces gènes est représentée par un paramètre d'entrée (voir Section 5.3).

Le graphe \mathcal{G} d'interactions entre ces gènes est donné dans la Figure 6.3 où les gènes d'entrée sont représentés par des cercles en pointillés. Les interactions et les hypothèses de compositions de ces interactions sont données dans la Table 6.3. Ce modèle provient de [43].

espèces	interactions	hypothèses de compositions d'interactions
<i>gt</i>	$i_{gt}^1 = (hb_z, t_{hb_z}^1, gt)$ $i_{gt}^2 = (kr, t_{kr}^1, gt)$ $i_{gt}^3 = (bcd, t_{bcd}^1, gt)$ $i_{gt}^4 = (cad, t_{cad}^2, gt)$	$\{ (ao((\dot{-}i_{gt}^1)), 1),$ $(ao((\dot{-}i_{gt}^2)), 2),$ $(ao(i_{gt}^3), 3),$ $(ao(i_{gt}^4), 4) \}$
<i>hb_z</i>	$i_{hb_z}^1 = (kr, t_{kr}^2, hb_z)$ $i_{hb_z}^2 = (bcd, t_{bcd}^1, hb_z)$ $i_{hb_z}^3 = (bcd, t_{bcd}^2, hb_z)$ $i_{hb_z}^4 = (bcd, t_{bcd}^3, hb_z)$ $i_{hb_z}^5 = (hb_m, t_{hb_m}^1, hb_z)$	$\{ (ao((\dot{-}i_{hb_z}^1)), 1),$ $(ao(i_{hb_z}^2), 2),$ $(ao((i_{hb_z}^2 \wedge i_{hb_z}^5)), 3),$ $(ao((i_{hb_z}^3 \wedge i_{hb_z}^5)), 4),$ $(ao((i_{hb_z}^4 \wedge i_{hb_z}^5)), 5) \}$
<i>kr</i>	$i_{kr}^1 = (gt, t_{gt}^1, kr)$ $i_{kr}^2 = (hb_z, t_{hb_z}^1, kr)$ $i_{kr}^3 = (hb_z, t_{hb_z}^3, kr)$ $i_{kr}^4 = (kni, t_{kni}^1, kr)$ $i_{kr}^5 = (bcd, t_{bcd}^1, kr)$	$\{ (ao((\dot{-}i_{kr}^1)), 1),$ $(ao((i_{kr}^2 \wedge (\dot{-}i_{kr}^3))), 2),$ $(ao((\dot{-}i_{kr}^4)), 3),$ $(ao(i_{kr}^5), 4) \}$
<i>kni</i>	$i_{kni}^1 = (gt, t_{gt}^1, kni)$ $i_{kni}^2 = (hb_z, t_{hb_z}^2, kni)$ $i_{kni}^3 = (bcd, t_{bcd}^1, kni)$ $i_{kni}^4 = (cad, t_{cad}^1, kni)$	$\{ (ao((\dot{-}i_{kni}^1)), 1),$ $(ao((\dot{-}i_{kni}^2)), 2),$ $(ao(i_{kni}^3), 3),$ $(ao(i_{kni}^4), 4) \}$

TAB. 6.3 – Interactions et hypothèses sur leurs compositions pour le modèle du module *gap-gene* de la segmentation de l'embryon de drosophile.

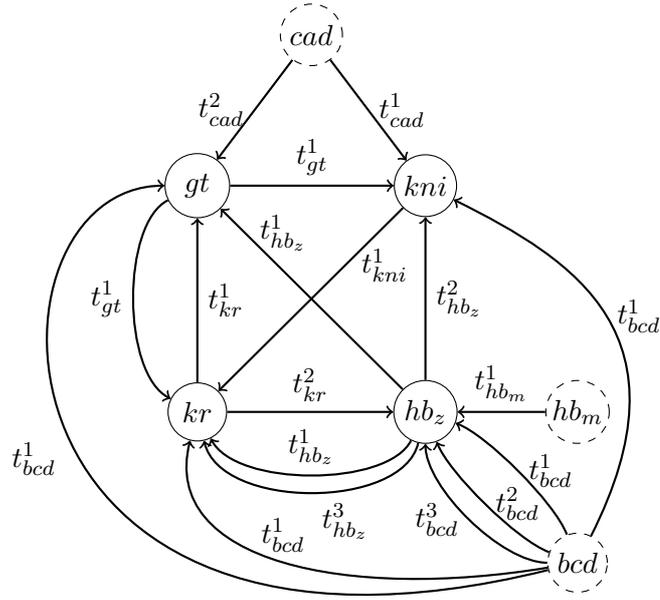


FIG. 6.3 – Graphe \mathcal{G} d'interaction pour le modèle du module *gap-gene* de la segmentation de l'embryon de drosophile.

L'embryon de drosophile est segmenté en plusieurs régions selon l'axe antérieur-postérieur. En [43] la modélisation prend en compte quatre régions adjacentes selon cet axe, noté A , B , C et D . Des expériences moléculaires ont produit des observations sur les concentrations des protéines impliquées dans le réseau pour le type sauvage, noté wt , et pour les neuf types mutants suivants : knock-out (KO) sur gt (la tendance de la concentration de gt est égal à 0 dans tout l'espace des états de concentration) noté $gt0$, KO à la fois sur hb_z et sur hb_m noté $hb0$, KO sur kr noté $kr0$, KO sur kni noté $kni0$, KO sur bcd noté $bcd0$, KO sur cad noté $cad0$, KO sur hb_m noté hb_m0 , expression ectopique égal à 1 sur gt (la tendance de la concentration de la protéine gt est partout égal à 1) noté $gt1$, expression ectopique égal à 1 sur kni noté $kni1$. Nous définissons, selon la méthode présentée dans la Section 5.2, un modèle M^T pour chacun des types $T \in \{wt, gt0, hb0, kr0, kni0, bcd0, cad0, hb_m0, gt1, kni1\}$. Par exemple, M^{gt0} correspond au modèle du type mutant $gt0$. Le lien entre les paramètres cinétiques discrets des différents modèles est établi comme présenté en Section 5.2.

Les observations concernent l'existence d'un état stationnaire par type et par région. Nous interprétons et formalisons les observations par l'existence d'un état stationnaire pour chaque type et chaque région, et par des contraintes entre ces états stationnaires. Les états stationnaires sont représentés par des listes/vecteurs des quatre concentrations de protéines produites par les gènes gt , hb_{zyg} , kr et kni noté $S^{R,T} = [S_{gt}^{R,T}, S_{hb_z}^{R,T}, S_{kr}^{R,T}, S_{kni}^{R,T}]$ pour chaque région $R \in \{A, B, C, D\}$ et chaque type mutant T . De la même manière les concentrations des protéines produites par les gènes d'entrée bcd , hb_m et cad pour chaque région R et chaque type mutant T sont notés respectivement $S_{bcd}^{R,T}$, $S_{hb_m}^{R,T}$ et $S_{cad}^{R,T}$.

La contrainte d'observation de l'état stationnaire pour chaque région R et chaque type T est :

$$\forall T, \forall R \\ S^{R,T} = [S_{gt}^{R,T}, S_{hb_z}^{R,T}, S_{kr}^{R,T}, S_{kni}^{R,T}] \wedge \\ path(M^T, [S^{R,T}, S^{R,T}], \mathcal{P})$$

Pour que la segmentation se déroule bien la mère offre, grâce aux concentrations des protéines produites par les gènes d'entrée, quatre contextes distincts. La contrainte d'observation de quatre entrées distinctes pour les types wt , $gt0$, $kr0$, $kni0$, $gt1$ et $kni1$ est :

$$\forall T \in \{wt, gt0, kr0, kni0, gt1, kni1\} \\ \forall (R1, R2) \in \{A, B, C, D\}^2, R1 \neq R2 \\ [S_{bcd}^{R1,T}, S_{hb_m}^{R1,T}, S_{cad}^{R1,T}] \neq [S_{bcd}^{R2,T}, S_{hb_m}^{R2,T}, S_{cad}^{R2,T}]$$

De plus il faut imposer que les entrées sont identiques entre différents types, sauf pour les entrées mutées :

$$\forall R \\ S_{bcd}^{R,hb0} = S_{bcd}^{R,wt}, \quad S_{cad}^{R,hb0} = S_{cad}^{R,wt} \\ S_{bcd}^{R,bcd0} = S_{bcd}^{R,wt}, \quad S_{cad}^{R,bcd0} = S_{cad}^{R,wt} \\ S_{hb_m}^{R,hb_m0} = S_{hb_m}^{R,wt}, \quad S_{cad}^{R,hb_m0} = S_{cad}^{R,wt} \\ S_{bcd}^{R,hb_m0} = S_{bcd}^{R,wt}, \quad S_{cad}^{R,hb_m0} = S_{cad}^{R,wt} \\ \forall T \in \{gt0, kr0, kni0, gt1, kni1\} \\ [S_{bcd}^{R,T}, S_{hb_m}^{R,T}, S_{cad}^{R,T}] = [S_{bcd}^{R,wt}, S_{hb_m}^{R,wt}, S_{cad}^{R,wt}]$$

La Table 6.4 donne les contraintes entre les concentrations discrètes des états stationnaires dans les régions A , B , C et D de chaque espèce et pour chaque type mutant ; le symbole \circ représente une absence de contrainte, et la fonte grasse indique une variation de contrainte par rapport au type sauvage. Toutes ces contraintes viennent de l'interprétation des données de [43]. La première colonne de la Table donne le type mutant, la seconde les gènes, la troisième représente l'état stationnaire de la région A , la quatrième, cinquième et sixième l'état stationnaire pour les régions B , C et D . Entre les troisième, quatrième, cinquième et sixième colonnes apparaissent des inégalités. Elles indiquent des contraintes entre les concentrations des états stationnaires de régions adjacentes. Finalement la septième colonne donne d'autres contraintes qui implique des concentrations pour des régions qui ne sont pas forcément adjacentes.

6.3.2 Nombre minimal de seuils

Dans les applications précédentes les seuils t_c^p sont égaux à p , p prenant une valeur entre 1 et max_c . Ceci implique que les concentrations de c peuvent prendre des valeurs entre 0 et max_c . Dans la mesure où le découpage de l'espace des concentrations discrètes n'est que spéculatif et que l'on ne connaît pas l'ordre des seuils de concentration il est intéressant de se demander si autant de seuils sont utiles pour satisfaire les observations. Le cas extrême est celui de la satisfaction de toutes les observations et hypothèses restreint au cas Booléen où toutes les concentrations sont comprises dans l'intervalle $[0, 1]$.

Il est facile grâce à notre outil de poser des contraintes sur les variables de seuils (voir Section 5.4) afin d'imposer des valeurs Booléennes aux concentrations. Il suffit pour cela de fixer tous les seuils t_c^p à 1, et donc fixer du même coup max_c à 1 pour toute espèce c .

Il apparaît que l'ensemble de toutes les contraintes, concernant les états stationnaires des 4 régions pour chacun des 10 types (sauvage et mutants), est inconsistant si nous imposons que

tous les seuils sont égaux à 1.

Afin d'identifier le nombre minimal de seuils nécessaires pour satisfaire toutes les hypothèses, il faut construire une requête utilisant le principe de relâchement de contraintes, présenté en Section 5.1. Mais cette fois ci, le relâchement a lieu sur le nombre de seuils de chaque espèce c en partant d'un nombre de seuils $Min = 1$ jusqu'à un nombre de seuils Max égal à la valeur maximale de p des t_c^p . Nous souhaitons d'autre part conserver tout de même les contraintes d'inégalités entre les seuils pour ce modèle de segmentation de la drosophile :

$$\forall p1, p2, (p1 < p2 \Rightarrow t_c^{p1} \leq t_c^{p2})$$

Pour faire le lien avec les Sections 5.4 et 5.1 disons que nous remettons en question les hypothèses de nombre de seuils pour toutes les espèces. Nous introduisons pour cela :

- deux Booléens B_1 et B_2 pour l'espèce hb_z (son nombre de seuils étant alors compris entre 1 et 3). B_1 est équivalent à "le nombre de seuils pour hb_z est strictement inférieur à 2". B_2 est équivalent à "le nombre de seuils pour hb_z est strictement inférieur à 3".
- un Booléen B_3 pour kr (définit de manière similaire aux Booléens B_1 et B_2).
- deux Booléens B_4 et B_5 pour bcd .
- un Booléen B_6 pour cad .

Finalement, nous cherchons à obtenir la prédiction complète $R1$ obtenue par disjonction de conditions de relâchement. Pour cette requête nous obtenons avec GNBox en 63 secondes la propriété nécessaire $R1$ suivante :

$$R1 \Leftrightarrow (\neg B_1 \wedge \neg B_2 \wedge \neg B_3 \wedge \neg B_4 \wedge \neg B_5 \wedge \neg B_6)$$

Cette requête indique donc que tous les seuils introduits dans [43] sont nécessaires vis-à-vis des hypothèses de modélisation et des observations faites sur le type sauvage et les neuf types mutants.

6.3. Application au module gap-gene de la segmentation de l'embryon de *Drosophila melanogaster*

type	espèce	A	B	C	D	contr. supp.
wild	bcd	$S_{bcd}^A \succ$	$S_{bcd}^B \succ$	$S_{bcd}^C \succ$	S_{bcd}^D	$S_{bcd}^D = 0$
	hbm	$S_{hbm}^A \succ$	$S_{hbm}^B \succ$	$S_{hbm}^C \succ$	S_{hbm}^D	$S_{hbm}^D = 0$
	cad	$S_{cad}^A \succ$	$S_{cad}^B \succ$	$S_{cad}^C \succ$	S_{cad}^D	$S_{cad}^D = 0$
	gt	$S_{gt}^A \succ$	$S_{gt}^B \circ$	$S_{gt}^C \prec$	S_{gt}^D	
	hbz	$S_{hbz}^A \succ$	$S_{hbz}^B \succ$	$S_{hbz}^C \succ$	S_{hbz}^D	$S_{hbz}^B > S_{hbz}^D$
	kr	$S_{kr}^A \prec$	$S_{kr}^B \prec$	$S_{kr}^C \prec$	S_{kr}^D	
kn1	$S_{kn1}^A \prec$	$S_{kn1}^B \prec$	$S_{kn1}^C \succ$	S_{kn1}^D		
gt0	bcd	$S_{bcd}^{A,gt0} \succ$	$S_{bcd}^{B,gt0} \succ$	$S_{bcd}^{C,gt0} \succ$	$S_{bcd}^{D,gt0}$	$S_{bcd}^{D,gt0} = 0$
	hbm	$S_{hbm}^{A,gt0} \succ$	$S_{hbm}^{B,gt0} \succ$	$S_{hbm}^{C,gt0} \succ$	$S_{hbm}^{D,gt0}$	$S_{hbm}^{D,gt0} = 0$
	cad	$S_{cad}^{A,gt0} \prec$	$S_{cad}^{B,gt0} \prec$	$S_{cad}^{C,gt0} \prec$	$S_{cad}^{D,gt0}$	$S_{cad}^{D,gt0} = 0$
	gt	$S_{gt}^{A,gt0} \circ$	$S_{gt}^{B,gt0} \circ$	$S_{gt}^{C,gt0} \circ$	$S_{gt}^{D,gt0}$	
	hbz	$S_{hbz}^{A,gt0} \succ$	$S_{hbz}^{B,gt0} \succ$	$S_{hbz}^{C,gt0} \succ$	$S_{hbz}^{D,gt0}$	$S_{hbz}^{B,gt0} > S_{hbz}^{D,gt0}$
	kr	$S_{kr}^{A,gt0} \prec$	$S_{kr}^{B,gt0} \prec$	$S_{kr}^{C,gt0} \prec$	$S_{kr}^{D,gt0}$	
kn1	$S_{kn1}^{A,gt0} \prec$	$S_{kn1}^{B,gt0} \prec$	$S_{kn1}^{C,gt0} \prec$	$S_{kn1}^{D,gt0}$		
hb0	bcd	$S_{bcd}^{A,hb0} \succ$	$S_{bcd}^{B,hb0} \succ$	$S_{bcd}^{C,hb0} \succ$	$S_{bcd}^{D,hb0}$	$S_{bcd}^{D,hb0} = 0$
	hbm	$S_{hbm}^{A,hb0} \circ$	$S_{hbm}^{B,hb0} \circ$	$S_{hbm}^{C,hb0} \circ$	$S_{hbm}^{D,hb0}$	$\forall R, S_{hbm}^{R,hb0} = 0$
	cad	$S_{cad}^{A,hb0} \prec$	$S_{cad}^{B,hb0} \prec$	$S_{cad}^{C,hb0} \prec$	$S_{cad}^{D,hb0}$	$S_{cad}^{D,hb0} = 0$
	gt	$S_{gt}^{A,hb0} \succ$	$S_{gt}^{B,hb0} \succ$	$S_{gt}^{C,hb0} \succ$	$S_{gt}^{D,hb0}$	$\forall R, S_{gt}^{R,hb0} \geq 1$
	hbz	$S_{hbz}^{A,hb0} \succ$	$S_{hbz}^{B,hb0} \succ$	$S_{hbz}^{C,hb0} \succ$	$S_{hbz}^{D,hb0}$	$S_{hbz}^{B,hb0} > S_{hbz}^{D,hb0}$
	kr	$S_{kr}^{A,hb0} \circ$	$S_{kr}^{B,hb0} \circ$	$S_{kr}^{C,hb0} \circ$	$S_{kr}^{D,hb0}$	$\forall R, S_{kr}^{R,hb0} = 0$
kn1	$S_{kn1}^{A,hb0} \circ$	$S_{kn1}^{B,hb0} \circ$	$S_{kn1}^{C,hb0} \circ$	$S_{kn1}^{D,hb0}$	$\forall R, S_{kn1}^{R,hb0} = 0$	
kr0	bcd	$S_{bcd}^{A,kr0} \succ$	$S_{bcd}^{B,kr0} \succ$	$S_{bcd}^{C,kr0} \succ$	$S_{bcd}^{D,kr0}$	$S_{bcd}^{D,kr0} = 0$
	hbm	$S_{hbm}^{A,kr0} \succ$	$S_{hbm}^{B,kr0} \succ$	$S_{hbm}^{C,kr0} \succ$	$S_{hbm}^{D,kr0}$	$S_{hbm}^{D,kr0} = 0$
	cad	$S_{cad}^{A,kr0} \prec$	$S_{cad}^{B,kr0} \prec$	$S_{cad}^{C,kr0} \prec$	$S_{cad}^{D,kr0}$	$S_{cad}^{D,kr0} = 0$
	gt	$S_{gt}^{A,kr0} \circ$	$S_{gt}^{B,kr0} \circ$	$S_{gt}^{C,kr0} \circ$	$S_{gt}^{D,kr0}$	$\forall R, S_{gt}^{R,kr0} \geq 1$
	hbz	$S_{hbz}^{A,kr0} \succ$	$S_{hbz}^{B,kr0} \succ$	$S_{hbz}^{C,kr0} \succ$	$S_{hbz}^{D,kr0}$	$S_{hbz}^{B,kr0} > S_{hbz}^{D,kr0}$
	kr	$S_{kr}^{A,kr0} \prec$	$S_{kr}^{B,kr0} \prec$	$S_{kr}^{C,kr0} \prec$	$S_{kr}^{D,kr0}$	
kn1	$S_{kn1}^{A,kr0} \circ$	$S_{kn1}^{B,kr0} \circ$	$S_{kn1}^{C,kr0} \circ$	$S_{kn1}^{D,kr0}$	$\forall R, S_{kn1}^{R,kr0} = 0$	
kn10	bcd	$S_{bcd}^{A,kn10} \succ$	$S_{bcd}^{B,kn10} \succ$	$S_{bcd}^{C,kn10} \succ$	$S_{bcd}^{D,kn10}$	$S_{bcd}^{D,kn10} = 0$
	hbm	$S_{hbm}^{A,kn10} \succ$	$S_{hbm}^{B,kn10} \succ$	$S_{hbm}^{C,kn10} \succ$	$S_{hbm}^{D,kn10}$	$S_{hbm}^{D,kn10} = 0$
	cad	$S_{cad}^{A,kn10} \prec$	$S_{cad}^{B,kn10} \prec$	$S_{cad}^{C,kn10} \prec$	$S_{cad}^{D,kn10}$	$S_{cad}^{D,kn10} = 0$
	gt	$S_{gt}^{A,kn10} \succ$	$S_{gt}^{B,kn10} \succ$	$S_{gt}^{C,kn10} \succ$	$S_{gt}^{D,kn10}$	
	hbz	$S_{hbz}^{A,kn10} \succ$	$S_{hbz}^{B,kn10} \succ$	$S_{hbz}^{C,kn10} \succ$	$S_{hbz}^{D,kn10}$	$S_{hbz}^{B,kn10} > S_{hbz}^{D,kn10}$
	kr	$S_{kr}^{A,kn10} \prec$	$S_{kr}^{B,kn10} \prec$	$S_{kr}^{C,kn10} \prec$	$S_{kr}^{D,kn10}$	$S_{kr}^{C,kn10} \geq 1$
kn1	$S_{kn1}^{A,kn10} \prec$	$S_{kn1}^{B,kn10} \prec$	$S_{kn1}^{C,kn10} \succ$	$S_{kn1}^{D,kn10}$		
bcd0	bcd	$S_{bcd}^{A,bcd0} \circ$	$S_{bcd}^{B,bcd0} \circ$	$S_{bcd}^{C,bcd0} \circ$	$S_{bcd}^{D,bcd0}$	$\forall R, S_{bcd}^{R,bcd0} = 0$
	hbm	$S_{hbm}^{A,bcd0} \succ$	$S_{hbm}^{B,bcd0} \succ$	$S_{hbm}^{C,bcd0} \succ$	$S_{hbm}^{D,bcd0}$	$S_{hbm}^{D,bcd0} = 0$
	cad	$S_{cad}^{A,bcd0} \prec$	$S_{cad}^{B,bcd0} \prec$	$S_{cad}^{C,bcd0} \prec$	$S_{cad}^{D,bcd0}$	$S_{cad}^{D,bcd0} = 0$
	gt	$S_{gt}^{A,bcd0} \succ$	$S_{gt}^{B,bcd0} \circ$	$S_{gt}^{C,bcd0} \prec$	$S_{gt}^{D,bcd0}$	$S_{gt}^{A,bcd0} = 0$
	hbz	$S_{hbz}^{A,bcd0} \circ$	$S_{hbz}^{B,bcd0} \circ$	$S_{hbz}^{C,bcd0} \circ$	$S_{hbz}^{D,bcd0}$	$\forall R, S_{hbz}^{R,bcd0} = 0$
	kr	$S_{kr}^{A,bcd0} \circ$	$S_{kr}^{B,bcd0} \circ$	$S_{kr}^{C,bcd0} \circ$	$S_{kr}^{D,bcd0}$	$\forall R, S_{kr}^{R,bcd0} = 0$
kn1	$S_{kn1}^{A,bcd0} \succ$	$S_{kn1}^{B,bcd0} \succ$	$S_{kn1}^{C,bcd0} \succ$	$S_{kn1}^{D,bcd0}$		
cad0	bcd	$S_{bcd}^{A,cad0} \succ$	$S_{bcd}^{B,cad0} \succ$	$S_{bcd}^{C,cad0} \succ$	$S_{bcd}^{D,cad0}$	$S_{bcd}^{D,cad0} = 0$
	hbm	$S_{hbm}^{A,cad0} \succ$	$S_{hbm}^{B,cad0} \succ$	$S_{hbm}^{C,cad0} \succ$	$S_{hbm}^{D,cad0}$	$S_{hbm}^{D,cad0} = 0$
	cad	$S_{cad}^{A,cad0} \circ$	$S_{cad}^{B,cad0} \circ$	$S_{cad}^{C,cad0} \circ$	$S_{cad}^{D,cad0}$	$\forall R, S_{cad}^{R,cad0} = 0$
	gt	$S_{gt}^{A,cad0} \succ$	$S_{gt}^{B,cad0} \circ$	$S_{gt}^{C,cad0} \circ$	$S_{gt}^{D,cad0}$	$S_{gt}^{D,cad0} = 0$
	hbz	$S_{hbz}^{A,cad0} \succ$	$S_{hbz}^{B,cad0} \succ$	$S_{hbz}^{C,cad0} \succ$	$S_{hbz}^{D,cad0}$	$S_{hbz}^{B,cad0} > S_{hbz}^{D,cad0}$
	kr	$S_{kr}^{A,cad0} \prec$	$S_{kr}^{B,cad0} \prec$	$S_{kr}^{C,cad0} \prec$	$S_{kr}^{D,cad0}$	$S_{kr}^{C,cad0} \geq 1$
kn1	$S_{kn1}^{A,cad0} \circ$	$S_{kn1}^{B,cad0} \circ$	$S_{kn1}^{C,cad0} \circ$	$S_{kn1}^{D,cad0}$	$\forall R, S_{kn1}^{R,cad0} = 0$	
hbm0	bcd	$S_{bcd}^{A,hbm0} \succ$	$S_{bcd}^{B,hbm0} \succ$	$S_{bcd}^{C,hbm0} \succ$	$S_{bcd}^{D,hbm0}$	$S_{bcd}^{D,hbm0} = 0$
	hbm	$S_{hbm}^{A,hbm0} \circ$	$S_{hbm}^{B,hbm0} \circ$	$S_{hbm}^{C,hbm0} \circ$	$S_{hbm}^{D,hbm0}$	$\forall R, S_{hbm}^{R,hbm0} = 0$
	cad	$S_{cad}^{A,hbm0} \prec$	$S_{cad}^{B,hbm0} \prec$	$S_{cad}^{C,hbm0} \prec$	$S_{cad}^{D,hbm0}$	$S_{cad}^{D,hbm0} = 0$
	gt	$S_{gt}^{A,hbm0} \succ$	$S_{gt}^{B,hbm0} \circ$	$S_{gt}^{C,hbm0} \prec$	$S_{gt}^{D,hbm0}$	
	hbz	$S_{hbz}^{A,hbm0} \succ$	$S_{hbz}^{B,hbm0} \succ$	$S_{hbz}^{C,hbm0} \succ$	$S_{hbz}^{D,hbm0}$	$S_{hbz}^{B,hbm0} > S_{hbz}^{D,hbm0}$
	kr	$S_{kr}^{A,hbm0} \prec$	$S_{kr}^{B,hbm0} \prec$	$S_{kr}^{C,hbm0} \prec$	$S_{kr}^{D,hbm0}$	
kn1	$S_{kn1}^{A,hbm0} \prec$	$S_{kn1}^{B,hbm0} \prec$	$S_{kn1}^{C,hbm0} \succ$	$S_{kn1}^{D,hbm0}$		
gt1	bcd	$S_{bcd}^{A,gt1} \succ$	$S_{bcd}^{B,gt1} \succ$	$S_{bcd}^{C,gt1} \succ$	$S_{bcd}^{D,gt1}$	$S_{bcd}^{D,gt1} = 0$
	hbm	$S_{hbm}^{A,gt1} \succ$	$S_{hbm}^{B,gt1} \succ$	$S_{hbm}^{C,gt1} \succ$	$S_{hbm}^{D,gt1}$	$S_{hbm}^{D,gt1} = 0$
	cad	$S_{cad}^{A,gt1} \prec$	$S_{cad}^{B,gt1} \prec$	$S_{cad}^{C,gt1} \prec$	$S_{cad}^{D,gt1}$	$S_{cad}^{D,gt1} = 0$
	gt	$S_{gt}^{A,gt1} \succ$	$S_{gt}^{B,gt1} \circ$	$S_{gt}^{C,gt1} \circ$	$S_{gt}^{D,gt1}$	
	hbz	$S_{hbz}^{A,gt1} \succ$	$S_{hbz}^{B,gt1} \succ$	$S_{hbz}^{C,gt1} \succ$	$S_{hbz}^{D,gt1}$	$S_{hbz}^{B,gt1} > S_{hbz}^{D,gt1}$
	kr	$S_{kr}^{A,gt1} \prec$	$S_{kr}^{B,gt1} \prec$	$S_{kr}^{C,gt1} \prec$	$S_{kr}^{D,gt1}$	
kn1	$S_{kn1}^{A,gt1} \circ$	$S_{kn1}^{B,gt1} \circ$	$S_{kn1}^{C,gt1} \circ$	$S_{kn1}^{D,gt1}$	$\forall R, S_{kn1}^{R,gt1} = 0$	
kn11	bcd	$S_{bcd}^{A,kn11} \succ$	$S_{bcd}^{B,kn11} \succ$	$S_{bcd}^{C,kn11} \succ$	$S_{bcd}^{D,kn11}$	$S_{bcd}^{D,kn11} = 0$
	hbm	$S_{hbm}^{A,kn11} \succ$	$S_{hbm}^{B,kn11} \succ$	$S_{hbm}^{C,kn11} \succ$	$S_{hbm}^{D,kn11}$	$S_{hbm}^{D,kn11} = 0$
	cad	$S_{cad}^{A,kn11} \prec$	$S_{cad}^{B,kn11} \prec$	$S_{cad}^{C,kn11} \prec$	$S_{cad}^{D,kn11}$	$S_{cad}^{D,kn11} = 0$
	gt	$S_{gt}^{A,kn11} \succ$	$S_{gt}^{B,kn11} \succ$	$S_{gt}^{C,kn11} \succ$	$S_{gt}^{D,kn11}$	
	hbz	$S_{hbz}^{A,kn11} \succ$	$S_{hbz}^{B,kn11} \succ$	$S_{hbz}^{C,kn11} \succ$	$S_{hbz}^{D,kn11}$	$S_{hbz}^{B,kn11} > S_{hbz}^{D,kn11}$
	kr	$S_{kr}^{A,kn11} \circ$	$S_{kr}^{B,kn11} \circ$	$S_{kr}^{C,kn11} \circ$	$S_{kr}^{D,kn11}$	$S_{kr}^{A,kn11} = S_{kr}^{B,kn11} = 0$
kn1	$S_{kn1}^{A,kn11} \prec$	$S_{kn1}^{B,kn11} \prec$	$S_{kn1}^{C,kn11} \succ$	$S_{kn1}^{D,kn11}$		

TAB. 6.4 – Contraintes entre les états stationnaires de chaque région pour chaque type mutant.

Implémentation

Cette partie discute de l'implémentation de l'outil GNBox. Le Chapitre 7 présente la mise œuvre de cet outil en PLC et en illustre la simplicité. Le Chapitre 8 présente l'intégration d'un solveur SAT dans GNBox afin d'en augmenter ses performances.

Chapitre 7

Mise en œuvre en PLC de GNBox

Le programme GNBox est réalisé dans l’environnement SICStus Prolog 4 et utilise son solveur sur domaines finis [11]. Cet environnement fait à l’heure actuelle référence dans le domaine de la Programmation Logique par Contraintes (PLC). Un programme en PLC est une formule logique définie à l’aide de prédicats fournis par une bibliothèque ou définis par l’utilisateur. GNBox est un outil composé de tels prédicats utilisateur. Les prédicats permettant la mise en œuvre de la construction d’un modèle \mathcal{M} à partir de \mathcal{G} sont présentés en Section 7.1, ceux permettant la construction de $transition(\mathcal{M}, ((S, S'), \mathcal{P}))$ et $path(\mathcal{M}, Path, \mathcal{P})$ en Section 7.2. En Section 7.3, nous présentons le prédicat permettant la mise en œuvre des extensions de fonctionnalités de GNBox (voir Chapitre 5) mais aussi des propriétés sur la manière de composer les interactions (voir Section 4.1). De plus GNBox assure la production automatique de prédictions complètes, sa mise en œuvre est présentée en Section 7.4. Enfin l’architecture générale des prédicats de GNBox est décrite en Section 7.5.

7.1 Mise en œuvre de la traduction d’un graphe \mathcal{G} en un modèle \mathcal{M}

Nous présentons ici les grandes lignes de cette mise en œuvre en utilisant les notions de prédicats de termes logiques¹⁷. Cela permet d’illustrer la simplicité du passage de la spécification à l’implémentation.

Un graphe \mathcal{G} est représenté par un terme G et un modèle \mathcal{M} par un terme M . Ces deux termes sont des listes. Le prédicat introduit est $trans_premodel(G, M)$ vrai si M est la traduction de G . Nous décrivons la structure de G en Sous-Section 7.1.1, celle de M en Sous-Section 7.1.2 et le prédicat $trans_premodel(G, M)$ en Sous-Section 7.1.3.

7.1.1 Représentation informatique d’un graphe \mathcal{G} d’interactions

Une liste G représentant un graphe \mathcal{G} d’interactions contient les représentations des noms des espèces, des arêtes de \mathcal{G} , des nombres maximaux de seuils de chaque espèce et des ordres connus entre les seuils. Les arêtes de \mathcal{G} sont représentées par une liste LI de termes T respectant la grammaire suivante :

¹⁷Nous utilisons bien sur des facilités de “compilation” évitant de reconstruire des termes, que nous ne décrivons pas ici

$$\begin{aligned}
 T & ::= c(C, LI_C) \\
 C & ::= \text{atome (identifiant de composante)} \\
 LI_C & ::= \text{liste de } I \\
 I & ::= i(C, P) \\
 P & ::= \text{constante entière positive (indice de seuil)}
 \end{aligned}$$

Une interaction $i_{C_1}^R = (C_2, t_{C_2}^P, C_1)$ d'un graphe \mathcal{G} est représentée par le R^{eme} élément $i(C_2, P)$ de LI_C1 , où LI_C1 apparaît dans le terme $T = c(C_1, LI_C1)$.

Exemple 7.1 Dans l'Annexe D.1 la liste LI représente la liste des interactions du graphe \mathcal{G} de la Table 3.1. Le deuxième élément de LI , $c(y, [i(x, 1), i(y, 2)])$, indique que deux interactions agissent sur la composante y . La première d'indice 1 est i_y^1 et vaut (x, t_x^1, y) étant donné que le premier élément de LI est le terme $i(x, 1)$. L'autre est $i_y^2 = (y, t_y^2, x)$ représentée par le terme $i(y, 2)$.

7.1.2 Représentation informatique d'un modèle

Un modèle est représenté par une liste $Model = [Species, Eqs, P, Nb_thr]$ où $Species$ est la liste des identifiants des espèces, Eqs un terme représentant le système d'équations focales, P une liste $[Input, Thr, K]$ où $Input$ est la liste des paramètres d'entrées, Thr la liste des paramètres de seuils, K la liste des paramètres cinétiques discrets et Nb_thr est la liste des nombres maximaux de seuils de chaque espèce.

Le terme Eqs est une liste de termes représentant chacun une équation focale Eq dont la grammaire est la suivante :

$$\begin{aligned}
 Eq & ::= eq(C, LP_C, LI_C, LV_C) \\
 C & ::= \text{atome (identifiant de composante)} \\
 LP_C & ::= \text{liste de } K \\
 K & ::= \text{variable entière positive (paramètre cinétique discret } K_C^L) \\
 LI_C & ::= \text{liste de } I \\
 I & ::= i(C, Th) \\
 Th & ::= \text{variable entière positive (seuil } t_c^p) \\
 LV_C & ::= \text{liste de } V \\
 V & ::= \text{liste de constantes Booléennes (vecteur Booléen associé à un contexte cellulaire)}
 \end{aligned}$$

Une équation est donc représentée par une liste LP_C de variables de paramètres cinétiques discrets et une liste LV_C des contextes cellulaires associés à ces paramètres. Les contextes cellulaires sont représentés par des vecteurs Booléens V comme spécifié en Section 3.1. Ces vecteurs sont associés à des sous-ensembles d'interactions de LI_C comme indiqué en Section 3.1.

Soit une équation focale $F_{c,S} = \sum_{l=1}^{l_c} K_c^l Cell_{c,S}^l$ de l'espèce c . Sa représentation $eq(c, LP_C, LI_C, LV_C)$ est telle que LP_C et LV_C sont de taille l_c et les éléments de LV_C (les vecteurs de contexte cellulaire) sont de même taille que LI_C (les interactions sur c). Cette représentation de l'équation est interprétée (d'un point de vue équation focale, cf Section 7.2) comme suit : $F_{c,S}$ vaut le l^{eme} élément de LP_C si la condition $Cell_{c,S}^l$ associé au vecteur de contexte cellulaire en l^{eme} position de LV_C est satisfaite.

Exemple 7.2 Dans l'Annexe D.2 la liste *Eqs* représente le système d'équation focale du modèle \mathcal{M} de la Table 3.1. Le deuxième élément de *Eqs* est¹⁸ :

$$\begin{aligned} eq(C, LP_C, LI_C, LV_C) = \\ eq(y, \\ [Ky_1, Ky_2, Ky_3, Ky_4], \\ [i(x, Tx_1), i(y, Ty_2)], \\ [[0, 0], [0, 1], [1, 0], [1, 1]]) \end{aligned}$$

Le troisième élément de *LP_C*, *Ky_3*, est la variable du paramètre cinétique discret de y d'indice 3 correspondant à la tendance de y dans les états S où la condition $Cellc_{y,S}^3 = I_{y,S}^1 \wedge \neg I_{y,S}^2 = S_x \geq t_x^1 \wedge S_y < t_y^2$ est vrai. Cela est dû au fait que le troisième élément de *LV* est $[1, 0]$.

Nous pouvons remarquer que la variable Tx_1 représente t_x^1 et que Ty_2 représente t_y^2 (voir introduction des variables de seuils en Section 5.4).

7.1.3 Le prédicat $trans_premodel(G, M)$

Le principal problème dans la définition du prédicat $trans_premodel(G, M)$ est celui d'établir la relation entre la liste *LI* représentant les arêtes de G et la liste *Eqs* représentant les équations focales. La solution se révèle très simple. Pour chaque espèce, la liste *LI_C* de M est déduite de celle de G et les listes *LP_C* et *LV_C* sont établies en fonction du nombre d'interactions.

Exemple 7.3 Pour l'application au stress nutritionnel carboné chez *E. coli* en Section 6.2, la liste *LI* possède un terme $T = c(crp, LI_C)$, où *LI_C* est la liste des représentations des 5 interactions influençant l'espèce *crp* :

$$LI_C = [i(us, 1), i(crp, 1), i(cya, 1), i(fis, 1), i(fis, 2)]$$

Les listes *LP_C* et *LV_C* du terme $eq(crp, LP_C, LI_C, LV_C)$ de *Eqs* sont de taille 2^5 . *LV_C* est la liste de toutes les listes possible de taille 5 de Booléens instanciés (il y en a bien 2^5) :

$$LV_C = [[0, 0, 0, 0, 0], [0, 0, 0, 0, 1], [0, 0, 0, 1, 0], \dots, [1, 1, 1, 1, 1]]$$

LP_C est la liste des variables entières représentant les paramètres K_{crp}^l compris dans l'intervalle $[0, max_{crp}]$.

7.2 Mise en œuvre de $transition(\mathcal{M}, ((S, S'), \mathcal{P}))$ et $path(\mathcal{M}, Path, \mathcal{P})$

La sémantique des réseaux de Thomas ou passage d'un modèle \mathcal{M} à une relation formelle entre l'ensemble \mathcal{T} des transitions de la dynamique des concentrations discrètes et l'ensemble \mathcal{P} des paramètres est formalisée en terme de contraintes dans la Section 3.2. Le module *dynamic_model* contient ces contraintes permettant d'imposer dans une requête qu'une transition partiellement connue existe pour un modèle donné contenu dans une variable *Model* créé à l'aide du prédicat *biomodel*.

L'interprétation du terme *Eqs* du modèle *Model* faite par le prédicat *focal_state* (du module *focal_state*) a déjà été donnée en Sous-Section 7.1.2.

¹⁸En ne considérant pas les variables identifiants statiquement les variables PLC pour l'intégration d'un solveur SAT à GNBox ; voir Chapitre 8

7.3 Mise en œuvre des extensions des fonctionnalités de GNBox

Cette mise en œuvre est réalisée grâce à un unique prédicat nommé *biomodel* qui possède plusieurs paramètres¹⁹ dont :

- *Model* : le modèle,
- *Mutant* : le type de mutant auquel *Model* correspond,
- *Modif_nb_thr* : le nombre de seuils différents de chaque espèce pour *Model*,
- *Comp* : les hypothèses de compositions d’interaction pour *Model*, et
- *Relax* : les possibilités de relâchement d’hypothèses autorisées pour *Model*.

Le prédicat *biomodel* est implémenté automatiquement par l’appel au prédicat *trans_premodel* avec comme argument le nom du fichier contenant la représentation du graphe \mathcal{G} d’interactions à utiliser, et utilise pour sa définition les prédicats fournis dans le module *biomodel_tool*. *trans_premodel* est donc le traducteur de \mathcal{G} en \mathcal{M} implémenté en suivant la formalisation donnée en Section 3.1.

Les sous-sections suivantes donnent les représentations de ces paramètres autres que celle de *Model* (présentée en Sous-Section 7.1.2).

7.3.1 Représentation de *Mutant*

Le paramètre *Mutant* permet de fixer le type de mutant auquel *Model* correspond. La grammaire de *Mutant* est :

$$\begin{aligned}
 \textit{Mutant} & ::= \text{liste de } EE \\
 EE & ::= ee(C, Val) \\
 C & ::= \text{atome (identifiant de composante)} \\
 Val & ::= \text{constante entière positive}
 \end{aligned}$$

Cette représentation des mutations est interprétée par rapport au type sauvage comme suit : pour tout élément $ee(C, Val)$ de *Mutant*, tous les paramètres cinétiques discrets pour la composante C du modèle *Model* sont égaux à Val .

Exemple 7.4 Pour l’application de la Section 6.3 nous avons du créer des modèles pour le type sauvage *wt*, le type mutant *gt0* dû à un KO sur *gt* et le type mutant *kni1* dû à une expression ectopique sur *kni* de valeur 1. Ces trois modèles sont créés par un appel à *biomodel* où le paramètre *Mutant* est égal respectivement à [], [$ee(gt, 0)$] et [$ee(kni, 1)$].

7.3.2 Représentation de *Modif_nb_thr*

Le paramètre *Modif_nb_thr* permet de fixer le nombre de seuils d’une ou de plusieurs espèces. La grammaire de *Modif_nb_thr* est :

$$\begin{aligned}
 \textit{Modif_nb_thr} & ::= \text{liste de } NT \\
 NT & ::= nb_thr(C, Val) \\
 C & ::= \text{atome (identifiant de composante)} \\
 Val & ::= \text{constante entière positive}
 \end{aligned}$$

Cette représentation est interprétée comme suit : chaque élément $nb_thr(C, Val)$ indique que le modèle *Model* possède une contrainte spécifiant que le nombre de seuils différents de la composante C est égal à Val .

¹⁹Un autre paramètre, *Cnf*, est spécifique à l’intégration d’un solveur SAT à GNBox ; voir Chapitre 8

Exemple 7.5 Pour l'application de la Section 6.3 nous avons du créer des modèles dont tous les nombres de seuils différents sont égaux à 1 afin de créer des modèles Booléens. Ces modèles sont créés par un appel à `biomodel` où le paramètre `Modif_nb_thr` est égal à

$$[\text{nb_thr}(gt, 1), \text{nb_thr}(hb_zyg, 1), \text{nb_thr}(kr, 1), \text{nb_thr}(kni, 1), \\ \text{nb_thr}(bcd, 1), \text{nb_thr}(hb_mat, 1), \text{nb_thr}(cad, 1)]$$

7.3.3 Représentation de *Cplx*

Le paramètre *Cplx* permet d'ajouter des hypothèses sur les compositions d'interactions d'une ou de plusieurs espèces. La grammaire de *Cplx* est :

$$\begin{aligned} Cplx & ::= \text{liste de } CompC \\ CompC & ::= ci(C, LComp) \\ C & ::= \text{atome (identifiant de composante)} \\ LComp & ::= \text{liste de } HComp \\ HComp & ::= Comp \mid a(Comp) \mid o(Comp) \mid ao(Comp) \\ Comp & ::= SComp \mid and(Comp, Comp) \mid or(Comp, Comp) \\ SComp & ::= Ind \mid neg(Ind) \\ Ind & ::= \text{constante entière (indice d'interaction)} \end{aligned}$$

Cette représentation des hypothèses de composition d'interaction est interprétée comme suit : pour tout élément $ci(C, LComp)$ de *Cplx*, le modèle *Model* possède les contraintes de composition d'interactions indiqués par *LComp* pour l'espèce *C*. Le terme $and(Comp, Comp)$ s'interprète par $Comp \wedge Comp$, $or(Comp, Comp)$ par $Comp \vee Comp$, $neg(Comp)$ par $\neg Comp$, et *Ind* par l'interaction correspondante i_C^{Ind} .

Exemple 7.6 Pour l'application de la Section 6.3 il faut créer des modèles avec les hypothèses de composition d'interaction données dans la Table 6.3. Ces modèles sont créés par un appel à `biomodel` où le paramètre *Cplx* est égal à

$$[\text{ci}(gt, [\text{ao}(neg(1)), \text{ao}(neg(2)), \text{ao}(3), \text{ao}(4)]), \\ \text{ci}(hb_zyg, [\text{ao}(neg(1)), \text{ao}(2), \text{ao}(and(2, 5)), \text{ao}(and(3, 5)), \text{ao}(and(4, 5))]), \\ \text{ci}(kr, [\text{ao}(neg(1)), \text{ao}(and(2, neg(3))), \text{ao}(neg(4)), \text{ao}(5)]), \\ \text{ci}(kni, [\text{ao}(neg(1)), \text{ao}(neg(2)), \text{ao}(3), \text{ao}(4)])]$$

7.3.4 Représentation de *Relax*

Le paramètre *Relax* permet de créer les Booléens équivalents au relâchement des hypothèses que nous souhaitons remettre en question. Ce sont donc les Booléens $\neg C_i$, où les C_i sont les Booléens équivalent aux hypothèses de C^2 dans la Section 5.1. La grammaire de *Relax* est :

$$\begin{aligned} Relax & ::= \text{liste de } SRelax \\ SRelax & ::= \text{nb_thr}(C, LBR) \mid a(C, Ind, BR) \mid o(C, Ind, BR) \mid \text{supp}(BR) \\ C & ::= \text{identifiant de composante} \\ Ind & ::= \text{constante entière (indice de composition d'interaction)} \\ LBR & ::= \text{liste de } BR \\ BR & ::= \text{variables Booléennes} \end{aligned}$$

Cette représentation des possibilités de relâchement est interprétée comme suit :

- pour tout élément $a(C, Ind, BR)$, respectivement $o(C, Ind, BR)$, de *Relax* le modèle *Model* possède une contrainte spécifiant que la négation de la contrainte d’additivité, respectivement d’observabilité, de la Ind^{eme} composition d’interactions de l’espèce C définie dans *Cplexi* est équivalente à la variable Booléenne BR ,
- pour tout élément $nb_thr(C, LBR)$ de *Relax* le modèle *Model* possède une contrainte spécifiant que le i^{eme} élément de LBR est une variable Booléenne équivalente à “le nombre de seuils associé à l’espèce C est strictement inférieur à $m + i$, où m est le nombre de seuils imposé pour l’espèce C dans *Modif_nb_thr*”²⁰.

Exemple 7.7 Pour l’application de la Section 6.3 il faut créer des modèles dont les nombres de seuils sont égaux à 1 et permettre de relâcher tous les nombres de seuils. Ces modèles sont créés par un appel à *biomodel* où le paramètre *Modif_nb_thr* est tel que donné dans l’Exemple 7.5, tandis que *Relax* est égal à :

$$[\quad nb_thr(gt, _), nb_thr(hb_{zyg}, _), nb_thr(kr, _), nb_thr(kni, _), \\ nb_thr(bcd, _), nb_thr(hb_{mat}, _), nb_thr(cad, _) \quad]$$

Le symbole $_$ indique une variable anonyme que l’on ne souhaite pas contraindre en *SICStus Prolog*. Ces variables sont créées automatiquement par l’appel au prédicat *constraints_Relax* du module *biomodel*. Un exemple d’appel à ce prédicat est présent dans le fichier implémentant le prédicat *biomodel* pour le modèle de l’exemple de la Table 3.1 en Annexe D.2.

Le résultat de la requête de l’application de la Section 6.3 offre une seule solution de relâchement avec un seul relâchement du nombre de seuils pour l’espèce hb_z . Dans ce cas le nombre de seuils de hb_z est égal à 2 (et non à 3), le nombre de seuils pour cad est égal à 2, et toutes les autres espèces ont des concentrations Booléennes. La variable *Relax* est alors instanciée pour cette unique solution comme suit :

$$[\quad nb_thr(gt, []), nb_thr(hb_{zyg}, [0, 1]), nb_thr(kr, [1]), nb_thr(kni, []), \\ nb_thr(bcd, [1, 1]), nb_thr(hb_{mat}, []), nb_thr(cad, [0]) \quad]$$

7.4 Mise en œuvre de la production automatique de prédictions complètes

La mise en œuvre de la production automatique de prédictions complètes “minimales” repose sur une énumération dans l’ordre croissant ou décroissant de la somme des Booléens B_i égaux aux contraintes remises en question dans une requête (voir Section 5.1). Cela est dû au fait que nous souhaitons réduire au maximum $R1$ et $R2$ durant leur construction en utilisant la propriété $A \vee (A \wedge B) \Leftrightarrow A$. L’ordre d’énumération est alors très important afin d’obtenir les “petites” conditions (A) avant les plus “grosses” ($A \wedge B$), et de supprimer ces dernières à la construction si la condition A a déjà été obtenue. et permettant ainsi d’obtenir plus rapidement $R1$ et $R2$ que $R0$.

Le principe pour construire $R1$ est le suivant. GNBox crée une variable *Sum* égal à la somme des Booléens B_i . Il énumère *Sum* dans l’ordre décroissant, puis énumère les Booléens B_i et enfin énumère tout le reste des variables. Pour chaque solution trouvée GNBox identifie l’ensemble E des variables B_i instanciées à *faux* et ajoute une contrainte égal à la disjonction des Booléens de cet ensemble E . Cela évite donc bien de trouver une nouvelle solution ayant le même ensemble E . A la fin de l’énumération de toutes les solutions, GNBox définit $R1$ comme la disjonction de toutes

²⁰La taille de LBR est fixée vis-à-vis de m et du nombre maximal de seuils autorisé par \mathcal{G} pour l’espèce C

les conjonctions négatives associées à chacun des ensembles E trouvés, une conjonction négative associée à un ensemble E étant la conjonction des négations des Booléens B_i de l'ensemble E .

Exemple 7.8 *Ce principe appliqué à l'Exemple 5.2 pour construire R1 conduit à trouver une première solution pour $Sum = 4$ dont l'ensemble E est égal à $\{B_1, B_2\}$, à ajouter la contrainte $B_1 \vee B_2$, à continuer d'énumérer et à ne plus trouver de solution. Finalement, le seul ensemble E trouvé est $\{B_1, B_2\}$ permettant de définir R1 par $\neg B_1 \wedge \neg B_2$.*

Le principe pour construire R2 est similaire. GNBox crée une variable Sum égal à la somme des Booléens B_i . Il énumère Sum dans l'ordre décroissant, puis énumère les Booléens B_i et enfin énumère tout le reste des variables. Pour chaque solution trouvée GNBox identifie l'ensemble E des variables B_i instanciées à *vrai* et ajoute une contrainte égal à la disjonction des négations des Booléens de cet ensemble E . A la fin de l'énumération de toutes les solutions, GNBox définit R2 comme la disjonction de toutes les conjonctions positives associées à chacun des ensembles E trouvés, une conjonction positive associée à un ensemble E étant la conjonction des Booléens B_i de l'ensemble E .

Exemple 7.9 *Ce principe appliqué à l'Exemple 5.2 pour construire R2 conduit à trouver une première solution pour $Sum = 1$ dont l'ensemble E est égal à $\{B_4\}$, à ajouter la contrainte $\neg B_4$, à continuer d'énumérer, à trouver une deuxième solution pour $Sum = 3$ dont l'ensemble E est égal à $\{B_3, B_5, B_6\}$, à ajouter la contrainte $\neg B_3 \vee \neg B_5 \vee \neg B_6$, et à ne plus trouver de solution. Finalement, deux ensembles E ont été trouvés permettant de définir R2 par $(B_4) \vee (B_3 \wedge B_5 \wedge B_6)$.*

7.5 Architecture générale de GNBox

L'environnement SICStus Prolog permet d'organiser un programme en modules contenant chacun une partie des prédicats. Cette organisation sert bien entendu à obtenir une compilation plus efficace en indiquant les dépendances entre modules, mais permet surtout dans notre contexte de gérer un programme volumineux en programmation logique comme nous allons le voir dans ce Chapitre et qui ne cesse de s'étendre comme nous le verrons par exemple dans le Chapitre 8 (pour étendre le programme à l'utilisation d'un solveur SAT).

Les modules du programme en PLC, leur organisation et leurs liens sont donnés en Figure 7.1. Les trois modules à gauche (*biomodel_tool*, *biomodel* et *trans_premodel*) concernent l'implémentation du passage d'un graphe \mathcal{G} d'interactions à un modèle \mathcal{M} . Leur formalisation a été présentée en Section 3.1 et les détails de leur implémentation en PLC ont été donnés en Section 7.1. Les deux modules centraux (*focal_state* et *dynamic_model*) concernent l'implémentation du passage d'un modèle \mathcal{M} à une relation formelle entre l'ensemble \mathcal{T} des transitions et l'ensemble \mathcal{P} des paramètres relatifs à \mathcal{M} . Leur formalisation a été présentée en Section 3.2 et les détails de leur implémentation en PLC ont été données en Section 7.2. Le module *enumeration* contient les prédicats d'énumérations des solutions et solutions partielles (solutions réduites à un sous-ensemble des variables de la requête), et en particulier les heuristiques d'énumération pour construire des prédictions complètes (comme discuté en Sous-Section 7.4).

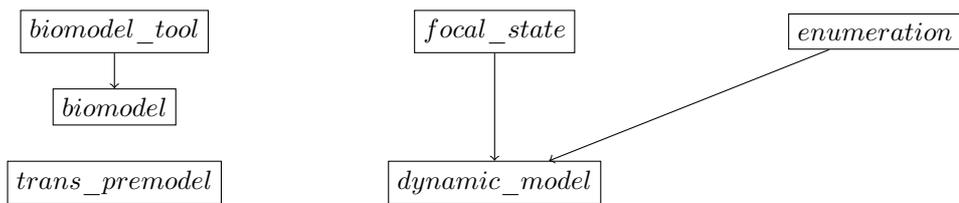


FIG. 7.1 – Architecture en modules du programme en PLC. Les nœuds du graphes sont les noms des modules. Une connexion orientée d’un module $m1$ à un module $m2$ signifie que des prédicats de $m1$ sont utilisés pour implémenter des prédicats de $m2$.

Chapitre 8

Intégration d'un solveur SAT à GNBox

Afin d'améliorer les performances de l'outil GNBox nous avons mis en œuvre et testé avec succès l'intégration d'un solveur SAT en conjonction avec le solveur sur domaines finis de SICS-tus Prolog 4 [19].

Un solveur SAT vérifie la satisfiabilité d'une formule en forme normale conjonctive (CNF). Une telle formule est une conjonction de *clauses*. Une clause est une disjonction de *littéraux* et un littéral est une variable Booléenne ou sa négation. Un solveur SAT permet d'établir une instantiation, s'il en existe une, de toutes les variables de la formule en CNF qui rend cette formule vraie. Une formule en CNF est *satisfiable* si elle accepte une telle instantiation, sinon elle est *insatisfiable*.

Exemple 8.1 *La formule CNF suivante (contenant 4 clauses et 3 variables) :*

$$(a \vee b) \wedge (\neg a \vee \neg c) \wedge (b) \wedge (a \vee \neg b \vee c)$$

est satisfiable car l'instanciation $a = \text{faux}, b = \text{vrai}, c = \text{vrai}$ rend cette formule vraie.

Les solveurs SAT sont optimisés pour trouver rapidement une solution à des formules dans ce format CNF très spécifique. L'efficacité des solveurs SAT, même sur des formules contenant des centaines de milliers de variables, est le résultat d'une grande adaptation au format très particulier des formules en CNF, d'un subtil équilibre entre déductions et énumérations et de l'utilisation du retour-arrière (*backtrack*) avec des heuristiques adaptatives [9].

Le problème clé pour résoudre efficacement un problème donné avec un solveur SAT est de trouver un encodage efficace en CNF. Les difficultés résident d'une part dans la détermination d'un ensemble de variables Booléennes de taille raisonnable traduisant l'ensemble des variables entières, d'autre part dans la construction des contraintes en taille et en nombre limité sur cet ensemble de variables Booléennes équivalentes aux contraintes sur les variables entières. Il se trouve que dans notre contexte la taille des domaines entiers est petite (dû à la discrétisation des concentrations par un faible nombre de seuils) et que les contraintes posées dans la définition des prédicats de notre programme sont aisément transformables en clauses sans explosion exponentielle de la taille des clauses ou de leur nombre. En Section 8.1 nous présentons la traduction d'une variable entière en un ensemble de variables Booléennes et un ensemble de clauses. En Section 8.2 nous décrivons rapidement la procédure de codage des contraintes 3.3, 3.2, 4.1, 4.2 et 4.3 en clauses et détaillons un exemple.

Il est important pour une utilisation aisée et étendue de l'outil GNBox de conserver la pose des contraintes et l'expression des requêtes en Prolog. En Section 8.3 nous donnons les grandes lignes de la mise en œuvre générale pour l'intégration d'un solveur SAT dans notre outil GNBox.

8.1 Représentation Booléenne d'une variable entière

Nous pouvons choisir un encodage très simple des variables entières dans la mesure où leur domaines sont très limités. Chaque valeur d'une variable entière donnée est codée par un Booléen. Ainsi, pour chaque variable entière de notre modèle nous introduisons autant de Booléens que la taille de son domaine et un ensemble de clauses spécifiant qu'exactly un de ces Booléens est vrai.

Exemple 8.2 Pour une variable X ayant comme domaine $\{0, 1, 2\}$, nous introduisons trois Booléens x_0, x_1, x_2 et les liens suivants entre X et ses valeurs : x_0 est équivalent à $X = 0$, x_1 est équivalent à $X = 1$, et x_2 est équivalent à $X = 2$. Il faut de plus ajouter les clauses spécifiant qu'un et un seul des Booléens est vrai : $x_0 \vee x_1 \vee x_2, \neg x_0 \vee \neg x_1, \neg x_0 \vee \neg x_2$ et $\neg x_1 \vee \neg x_2$.

Pour une variable avec un domaine de l valeurs nous obtenons une clause de l littéraux, et $l(l-1)/2$ clauses de 2 littéraux (qui donnent lieu à une propagation immédiate²¹). Nous obtenons finalement un ensemble de clauses d'une taille non pénalisante en introduisant les variables multivaluées avec le codage simple présenté. Pour toutes les autres variables (qui sont nécessairement Booléennes) nous ajoutons une unique variable Booléenne (et non deux variables Booléennes).

Par contre ce codage n'est pas intéressant si l est grand. Un autre codage possible est celui proposé dans [30]. Ce codage est linéaire vis-à-vis de l . Plus précisément il introduit $l-1$ nouvelles variables Booléennes et construit $4 * l - 5$ clauses de deux ou trois littéraux. Ce codage devient donc intéressant si $(l(l-1)/2) + 1 \geq 4 * l - 5$, et ceci est vrai pour $l \geq 8$. La présence de domaines de taille 8 ou plus implique une discrétisation des concentrations par au moins 7 seuils. Ce cas semble très rare dans les réseaux que nous étudions.

8.2 Traduction de relations entières en relations Booléennes

Il s'agit ici d'encoder les relations des Contraintes 3.3, 3.2, 4.1, 4.2 et 4.3. Elles concernent au plus deux variables multivaluées possédant des domaines de petite taille et potentiellement d'autres variables Booléennes. Ainsi *a priori* nous pouvons nous attendre à des formulations Booléennes peu lourdes pour ce type de relations entières. Néanmoins, il n'est pas raisonnable d'utiliser la méthode triviale de développement qui obtient la forme clausale à partir de la table de vérité de la relation initiale. En effet, dans cette optique nous aurions tout de même à faire face à une explosion exponentielle du nombre de clauses générées selon la taille des domaines des variables entières. Notre méthode évite cette explosion en prenant en compte le fait qu'une et une seule des variables Booléennes pour chaque valeur possibles d'une variable entière est vraie. Nous détaillons cette approche sur un exemple de traduction en forme clausale d'une relation entière typique.

Considérons la relation $B \Leftrightarrow X = Y$ (comme par exemple la relation $LBup_c \Leftrightarrow S'_c = S_c + 1$ que l'on trouve dans la Contrainte 3.3), avec les domaines $\{0, 1\}, \{0, 1, 2\} \{1, 2, 3\}$ pour B, X et Y respectivement. Soit $b, x_0, x_1, x_2, y_1, y_2$ et y_3 les variables Booléennes représentant les variables B, X et Y . Pour encoder cette relation, nous introduisons six clauses traduisant la conjonction des deux formules $B \Rightarrow (X = Y)$ et $B \Leftarrow (X = Y)$:

²¹Les solveurs SAT actuels équilibrent la propagation et l'énumération en ne propageant que sur les deux littéraux extérieurs de chaque clause. Ils évitent ainsi d'inspecter tous les littéraux de la clause et ne propagent donc pas au maximum. Par conséquent les clauses de taille 2 sont rapidement prises en compte. Pour ces clauses les solveurs propagent sur tous les littéraux de la clause.

- $B \Rightarrow (X = Y) : (\neg b \vee \neg x1 \vee y1) \wedge (\neg b \vee x1 \vee \neg y1) \wedge (\neg b \vee \neg x2 \vee y2) \wedge (\neg b \vee x2 \vee \neg y2)$
- $(X = Y) \Rightarrow B : (b \vee \neg x1 \vee \neg y1) \wedge (b \vee \neg x2 \vee \neg y2)$

Cet exemple donne une bonne idée de l'algorithme d'encodage de relations du type $B \Leftrightarrow X = Y$. L'algorithme boucle sur les valeurs de B , puis sur celles de X et finalement sur celles de Y , et produit les implications (clauses) sur les valeurs de ces différentes variables.

Notre compilateur traduit selon le même principe sous forme clausale tous les types de relations entières utiles dans notre modèle : $B1 \vee B2$, $B1 \wedge B2$, $B1 \Leftrightarrow B2$, $X = Y$, $X \neq Y$, $X \leq Y$, $X < Y$, $B \Rightarrow X = Y$, $X = Y \Rightarrow B$, $B \Rightarrow X < Y$, $X < Y \Rightarrow B$, $B \Leftrightarrow \bigwedge_i B_i$ et *exactly_one*($\langle \text{list of } B_i \rangle$), où les variables B sont des variables Booléennes et les variables X et Y sont des variables multivaluées. La taille des représentations en CNF de toutes ces relations est linéaire selon la taille des domaines des variables entières X et Y , excepté pour la relation de type *exactly_one*($\langle \text{list of } B_i \rangle$). La traduction de cette dernière produit certes un nombre quadratique de clauses selon le nombre d'espèces, mais avec deux littéraux seulement par clause.

Notre compilateur traduit aussi sous forme clausale d'autres types de relations entières non utiles dans notre modèle mais qui le sont dans les requêtes. Ces relations entières peuvent potentiellement porter sur plus de deux variables entières (par exemple $S_{gyr}^{ns} - S_{gyr}^s > S_{top}^{ns} - S_{top}^s$ que nous trouvons dans la formulation des hypothèses de la Sous-Section 6.2.1). Il est important de noter qu'une relation difficilement traduisible en forme clausale peut être traitée avec GNBox sans la traduire en forme clausale. C'est le solveur CLP qui s'occupe alors de la satisfiabilité de cette relation.

8.3 Coopération du solveur SAT et du solveur de contraintes PLC dans l'outil GNBox

La mise en œuvre de l'intégration d'un solveur SAT²² dans l'outil GNBox implémenté en SICStus Prolog repose sur :

- une traduction du problème, c'est-à-dire un réseau de Thomas, sur sous la forme de contraintes en un problème sur Booléens sous la forme d'un ensemble de clauses,
- une interface du solveur PLC vers le solveur SAT permettant dans une requête Prolog d'ajouter des clauses (provenant de la compilation du problème d'origine ou d'informations de propagation faites par le solveur PLC) et de lancer une résolution SAT de l'ensemble de clauses,
- une interface du solveur SAT vers le solveur PLC permettant de "récupérer" une solution SAT en contraignant selon cette solution les variables utilisées par le solveur PLC. De cette façon il est possible d'utiliser GNBox en n'utilisant que des prédicats et des requêtes Prolog.

La traduction en CNF est invisible pour l'utilisateur qui pose sa requête dans la ligne de commande Prolog. Par contre la forme clausale traduite du problème posé est accessible, en terme de variables Prolog, dans la version courante du logiciel, pour permettre une conception et un débogage plus facile. A cette fin chaque variable entière X de la formalisation est associée à un identificateur unique, qui est soit fixé à l'avance soit fixé à la traduction en CNF, et à un ensemble d'identificateurs uniques pour les variables SAT correspondantes à la variable X . Ce lien entre identificateurs permet d'utiliser deux fois (ou plus) la même variable PLC dans une requête, sans avoir à utiliser deux variables ayant des identificateurs distincts et à imposer

²²Nous utilisons le solveur SAT minisat implémenté en C et C++ [28], mais ce choix n'est pas fixé, car on peut s'attendre à des améliorations dans ce secteur très étudié.

qu'elles sont égales à la fois en CLP et en SAT. De plus, ce lien est nécessaire pour réaliser les interfaces entre les solveurs PLC et SAT, afin de transférer toute information sur les variables utilisées par les différents solveurs.

L'organisation et les liens de la majorité des modules de GNBox présentés dans le Chapitre 7²³ et ceux concernant l'utilisation d'un solveur SAT est donnée en Figure 8.1.

Le module *cnf* est le module le plus fondamental. Il traduit un ensemble de contraintes sur entiers en PLC en un ensemble de clauses. Pour cela il utilise des prédicats pour créer des identifiants de variables SAT, pour créer les clauses spécifiant qu'exactly un de ces Booléens est vrai (voir Section 8.1), et pour créer les clauses équivalentes aux contraintes sur entiers du programme et des requêtes en PLC (voir Section 8.2). L'Annexe G présente les aspects techniques liés à cette traduction.

Le module *sp_minisat* spécifie l'interface Prolog vers les fonctions C ou C++ permettant d'ajouter des clauses au solveur et de résoudre la satisfiabilité de l'ensemble de ces clauses. De plus, la résolution peut non seulement demander toutes les solutions (valuations de toutes les variables du problème) satisfaisant les contraintes, mais aussi toutes les solutions partielles (valuations d'un sous-ensemble donné de variables d'intérêt) satisfaisant les contraintes. Cette dernière subtilité permet entre autres de déterminer toutes les valuations de paramètres acceptant un ensemble de contraintes comportementales, ou de déterminer tous les états stationnaires.

Exemple 8.3 Soient les variables SAT suivantes : x_1 , x_2 et x_3 . Soit l'ensemble de trois clauses suivantes : $\{(x_2 \vee \neg x_1), (x_3 \vee \neg x_1), (x_1 \vee x_2)\}$. Cet ensemble de clauses est satisfiable pour trois valuations des variables : $(\neg x_1 \wedge x_2 \wedge \neg x_3)$, $(\neg x_1 \wedge x_2 \wedge x_3)$ et $(x_1 \wedge x_2 \wedge x_3)$. L'ensemble des solutions partielles pour le sous-ensemble de variables $\{x_1, x_3\}$ est $\{(\neg x_1 \wedge \neg x_3), (\neg x_1 \wedge x_3), (x_1 \wedge x_3)\}$. L'ensemble des solutions partielles pour le sous-ensemble de variables $\{x_1, x_2\}$ est $\{(\neg x_1 \wedge x_2), (x_1 \wedge x_2)\}$. L'ensemble des solutions partielles pour le sous-ensemble de variables $\{x_2\}$ est $\{(x_2)\}$.

Le module *interface_CP_SAT* permet principalement de transférer une solution SAT sur les variables PLC de la requête permettant ainsi une manipulation aisée des solutions dans la ligne de commande Prolog de la solution. Cette fonctionnalité permet non seulement d'implémenter le relâchement de contraintes et l'obtention de prédictions complètes au niveau Prolog, mais aussi de tester la satisfiabilité d'un ensemble de contraintes dont certaines ne sont pas traduites en clauses. C'est le solveur CLP qui en est chargé à partir des solutions SAT sur un sous-ensemble de l'ensemble des contraintes.

Le module *constraints_CP_SAT* fournit les prédicats permettant de créer des variables et de poser des contraintes sur ces variables à la fois pour le solveur PLC et pour le solveur SAT.

Le module *enumeration* est complété par des prédicats d'énumération de solutions et solutions partielles en utilisant non pas le solveur PLC mais le solveur SAT. L'énumération avec le solveur SAT des solutions partielles relatives à un sous-ensemble E (E pouvant être l'ensemble complet des variables) de variables est réalisée en ajoutant une *clause bloquante* relative à E à l'ensemble des clauses courantes après l'obtention de chaque solution S . Une clause bloquante

²³sauf *trans_premodel* et *reverse_interaction* qui ne sont pas des modules liés à l'utilisation d'un solveur SAT.

relative au sous-ensemble de variables E pour une solution S est la négation de la solution S réduite aux variables de E . Le rajout de cette clause bloquante évite d'obtenir de nouvelles solutions ayant la même solution partielle.

Exemple 8.4 Montrons comment, à l'aide d'énumération avec le solveur SAT, nous obtenons le sous-ensemble de solutions relatives au sous-ensemble de variables $\{x_1, x_2\}$ pour l'Exemple 8.3.

Imaginons que nous obtenons comme première solution $(\neg x_1 \wedge x_2 \wedge \neg x_3)$. La solution partielle relative à $\{x_1, x_2\}$ correspondant à cette solution est $(\neg x_1 \wedge x_2)$. La clause bloquante relative à $\{x_1, x_2\}$ pour cette solution est $(x_1 \vee \neg x_2)$. Après ajout de cette clause bloquante le solveur SAT détermine la solution $(x_1 \wedge x_2 \wedge x_3)$. La solution partielle relative à $\{x_1, x_2\}$ correspondant à cette solution est $(x_1 \wedge x_2)$. La clause bloquante relative à $\{x_1, x_2\}$ pour cette solution est $(\neg x_1 \vee \neg x_2)$. Après ajout de cette clause bloquante le solveur SAT ne trouve plus de solution. Ainsi nous obtenons bien les deux solutions partielles souhaitées (sans avoir énuméré toutes les solutions).

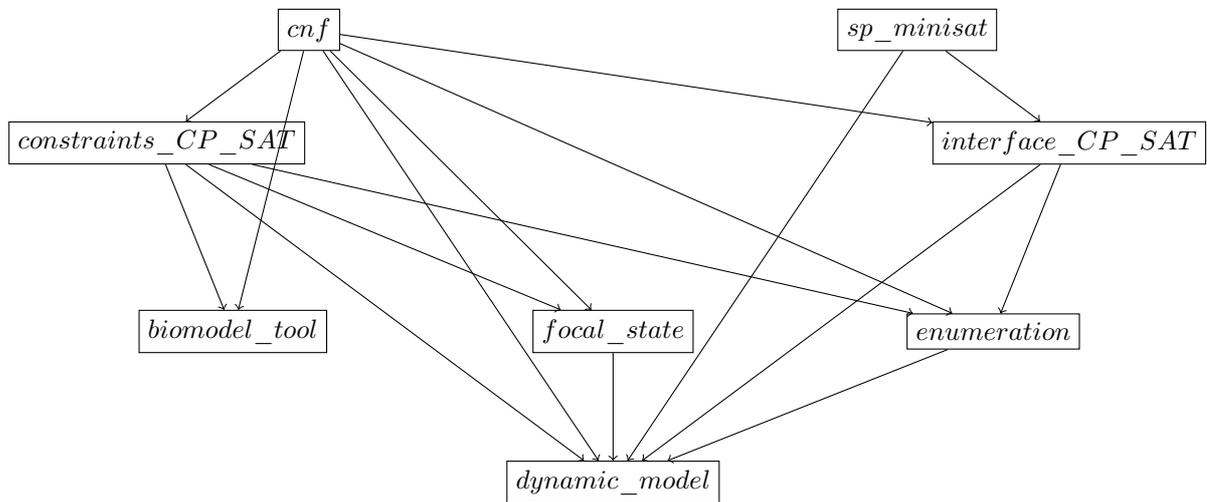


FIG. 8.1 – Architecture en modules du programme GNBox en PLC dont la partie concernant l'utilisation d'un solveur SAT (sauf les modules *outils* et *print* qui ne sont pas des modules fondamentaux, et *trans_premodel* et *reverse_interaction* qui ne sont pas des modules liés à l'utilisation d'un solveur SAT). Les nœuds du graphes sont les noms des modules. Une connexion orientée d'un module m_1 à un module m_2 signifie que des prédicats de m_1 sont utilisés pour implémenter des prédicats de m_2 .

8.4 Performances de l'outil GNBox

En l'absence d'un banc de tests composé de réseaux et de requêtes il est difficile d'apprécier d'une façon globale les performances de GNBox. Néanmoins les trois applications que nous avons étudiées semblent très illustratives. Elles concernent les réseaux décrits en Section 6.1 (4 gènes et 10 interactions), en Section 6.2 (5 gènes, une entrée et 19 interactions) et en Section 6.3 (4 gènes et 18 interactions).

Les performances de GNBox nous semblent pouvoir être appréciées suivant les trois types de requêtes : cohérence, relâchement en cas d'inconsistance, et prédiction de propriétés.

8.4.1 Recherche de cohérence. Performances

Concernant le réseau de la Section 6.1 la cohérence est établie sans coût significatif : 6 secondes, alors que la taille des chemins (48 états) considérée est relativement importante. Il est à signaler que le problème hautement combinatoire de la recherche du diamètre (43 états) de la classe des modèles cohérents est résolu en deux requêtes. L'une prouvant l'existence d'une solution pour une taille de 43 en 135 secondes. L'autre prouvant l'inconsistance pour une taille de 44 en 977 secondes. Comme ce type de requêtes n'est posé qu'une seule fois préalablement aux requêtes portant sur des comportements, ce temps de calcul de plus d'un quart d'heure paraît acceptable.

Pour le réseau de la Section 6.2 l'obtention de l'incohérence s'établit aussi rapidement : 6 secondes avec une taille de chemin de 10 (dont 1 seconde de résolution), 176 secondes avec une taille de chemin de 100 (dont 28 secondes de résolution). Notons qu'une taille maximum de chemin de 100 apparaît biologiquement très raisonnable. Si les chemins d'atteignabilité entre états perturbés et états stationnaires ne sont pas imposés une seule seconde suffit pour obtenir l'incohérence. Mais pour poser cette requête il faut savoir *a priori* que l'incohérence ne provient pas de la présence de ces chemins.

8.4.2 Relâchement. Performances

L'ensemble de contraintes décrivant le réseau de la Section 6.2 est incohérent. Le coût du relâchement qui propose d'éliminer deux contraintes reste négligeable si la taille des chemins est 10 : 55 secondes. Ce coût est nettement moins négligeable, bien que supportable, pour un taille de chemin de 100 : 1380 secondes, 23 minutes.

Le relâchement de contraintes sur le nombre de seuils différents est l'objectif principal de l'étude du réseau en Section 6.3. Ici le temps de calcul pour déterminer qu'effectivement tous les seuils doivent être différents n'est pas négligeable mais reste acceptable : 63 secondes. Le phénomène s'explique par la combinatoire importante produite par la nécessité de prouver l'inconsistance de sept classes de modèles de taille très importantes : ceux pour lesquels tous les seuils sont égaux, ceux pour lesquels une seule différence entre seuils apparaît, etc.

8.4.3 Prédications de propriétés. Performances

Les prédictions obtenues à partir du réseau de la Section 6.1 en l'occurrence une conjonction d'observabilité le sont rapidement : 5 secondes.

Il en est de même pour les prédictions, en l'occurrence une disjonction d'observabilité, obtenues à partir du réseau de la Section 6.2 : 14 secondes avec une taille de chemin de 10, et 362 secondes avec une taille de chemin de 100. Nous notons que les prédictions obtenues sont les mêmes quelque soient les tailles de chemins (10 et 100). Mais ce résultat n'est obtenue que *a posteriori*.

Conclusion

Chapitre 9

Conclusion

Cette thèse montre l'intérêt et la faisabilité à l'aide de la technologie par contraintes de l'approche déclarative pour la construction et l'analyse de réseaux de régulation biologiques discrets.

D'un **point de vue conceptuel**, nous avons représenté les formalisations sous forme de contraintes, qui plus est "exécutables", suivantes :

- formalisation d'un graphe \mathcal{G} d'interactions,
- formalisation sous forme de contraintes d'un modèle \mathcal{M} ,
- formalisation de la notion de composition d'interactions,
- formalisation de comportements dynamiques, en notant que la comparaison de la PLC avec des logiques temporelles comme CTL est favorable à la PLC dans notre contexte,

Il est à noter que la formalisation avec une approche déclarative à l'aide de contraintes exige une grande rigueur (exemple : notion d'observabilité).

D'un **point de vue mise en œuvre**, l'approche par contraintes repose et se distingue essentiellement, dans le cas où les paramètres ne sont pas fixés, sur une vision "représentation en intention" opposée à une vision "représentation en extension". L'outil GNBox assure de multiples fonctionnalités dont :

- la cohérence d'une ensemble d'hypothèses, qui est par défaut introduite avec la PLC,
- le relâchement automatique d'hypothèses dans le cas d'une incohérence et plus généralement l'obtention de prédictions complètes, qui sont des fonctionnalités nécessaires et facilement introduites grâce à la PLC

où les hypothèses sont exprimées directement par des contraintes sur les variables du problème (variables de concentration des chemins, paramètres cinétiques discrets, paramètres de seuils, paramètres d'entrée) ou en utilisant les facilités d'expression de l'outil GNBox concernant la définition de mutants et les propriétés de compositions d'interactions.

D'un **point de vue validation**, GNBox montre des performances très acceptables sur les requêtes des trois applications biologiques traitées grâce à l'intégration d'un solveur SAT, alors que cela était un enjeu délicat au départ.

Perspectives

- A partir des travaux déjà réalisés, plusieurs voies de recherche sont possibles :
- l’introduction des états singuliers pour prendre en compte des comportements glissants sur les seuils de discrétisation (voir Chapitre 2). Cette introduction des états singuliers complique les règles de calcul des successeurs mais nous avons montré dans d’autres travaux [18, 23] que ces règles peuvent être formalisées en PLC. Les états singuliers sont pris en compte dans une version plus légère de notre outil.
 - l’ajout de contraintes redondantes relatives aux conditions nécessaires de multistationarité et stabilité sur les circuits de \mathcal{G} sur la base des travaux [51, 45, 46, 53, 52, 15] dont une application est par exemple d’imposer efficacement des comportements cycliques [48].
 - l’introduction dans le formalisme de délais entre les ordres de productions des protéines et les débuts effectifs de ces productions. De tels délais permettent d’affiner l’abstraction dans le sens où certains chemins du graphe de transitions peuvent être éliminés. Les différents travaux introduisant des délais dans les réseaux génétiques pour faire des simulations et obtenir des contraintes sur les paramètres des délais afin d’autoriser un comportement spécifique utilisent : (i) soit une horloge continue évoluant dans un état discret et permettant [2, 3] ou non [44] de faire évoluer de manière non-monotone les taux de production et de dégradation des protéines le long des trajectoires, ou encore d’imposer que les taux de dégradation sont uniformes [1], (ii) soit un découpage plus fin des états de tel sorte que le signe des dérivés des variables de concentrations dans chaque état soit constant et donc sans introduire d’horloge continue [7, 6], (iii) soit des réseaux Booléens évoluant selon un temps continu [34] inspirés de méthodes de climatologie et sismologie. L’introduction d’une horloge continue dans notre formalisme semble une bonne piste d’extension et induit l’utilisation d’un solveur sur domaines réels et une coopération de ce solveur avec le solveur sur domaine fini d’entiers.
 - l’étude dans notre formalisme de la composition de réseaux. L’idée sous-jacente de cette étude est d’inférer automatiquement des propriétés sur une composition de réseaux à partir des propriétés de ces réseaux [32].
 - l’étude de la faisabilité du *reverse-engineering* du graphe \mathcal{G} d’interactions comme étudié par J. Cohen en [17] avec notre formalisme. L’idée intuitive et simpliste est de définir un graphe \mathcal{G} complètement maillé dont les nœuds sont donnés, en limitant tout de même le nombre de seuils par espèces, puis de chercher le plus petit ensemble d’interactions permettant de vérifier les hypothèses.

Annexes

Annexe A

Description minimaliste de la Programmation Logique par Contraintes (PLC)

La Programmation Logique par Contraintes (PLC) appartient au type de programmation dit “logique” ou déclaratif, dont l’objectif est de décrire le “quoi” plutôt que le “comment”.

Les langages impératifs sont les plus courants (exemple : FORTRAN, ALGOL, BASIC, COBOL, C, ADA, Java) et décrivent le “comment” en permettant de décrire les opérations à effectuer en termes d’états du programme. Pour cela les variables du programme sont liées à des valeurs (instanciées) avant d’être utilisées, et peuvent aussi changer de valeur.

Les langages déclaratifs décrivent le “quoi”. Ils regroupent les langages descriptifs (comme HTML, XML, LaTeX), les langages fonctionnels (comme LISP, Caml, Haskell, F#) et les langages logiques (comme Prolog ou Oz). Les langages logiques décrivent un programme à l’aide d’un ensemble de faits élémentaires et de formules logiques (la théorie). Un solveur (ou interpréteur ou démonstrateur de théorème) est chargé de répondre aux requêtes de l’utilisateur s’exprimant sous la forme de formules à démontrer. A ce paradigme logique il est naturel des relations spécifiques (les contraintes) entre plusieurs variables, nous obtenons alors les langages de PLC. Contrairement au paradigme impératif, ces variables ne sont pas instanciées et appartiennent à un domaine de valeur. Le solveur est chargé de réduire les domaines afin qu’ils satisfassent les contraintes spécifiées.

Dans certains cas il est possible que l’exécution conduise à ce que tous les domaines soient réduits à une seule valeur. Dans un tel cas la valuation obtenue est une solution à l’ensemble de contraintes initial. Afin de trouver les solutions l’utilisateur peut demander au solveur PLC d’énumérer toutes les solutions possibles. Cette énumération peut être coûteuse dans le cas où les domaines sont vastes.

Dans d’autres cas il est possible que l’exécution conduise à ce qu’un des domaines soit réduit à un ensemble vide. Dans un tel cas l’ensemble de contraintes est incohérent (ou inconsistant). Si cet exécution résulte d’une énumération le solveur se charge de déterminer si d’autres valuations donnent des solutions.

Nous distinguons deux aspects importants des langages en PLC. Le premier est déclaratif dans le sens où l’ordre de présentation n’a pas de rapport avec l’exécution. Ainsi un programme garde une interprétation logique permettant de garantir sa correction. Le second est opérationnel dans le sens où un ordinateur procède séquentiellement (usuellement) pour traiter la partie déclarative,

ainsi le programmeur peut exploiter les propriétés algorithmiques du solveur afin d'améliorer les performances.

A.1 Aspect déclaratif

La PLC permet à l'utilisateur de modéliser un problème par un ensemble de contraintes ou équations mathématiques sur un ensemble de variables. La PLC permet donc de modéliser des informations partielles. Cette expression du problème correspond à la partie déclarative de la requête. Il y a plusieurs types de langages PLC dépendant des types de domaines des variables. Par exemple les domaines peuvent inclure des réels, des Booléens ou être juste spécifiés comme fini (dans un tel cas les valeurs des variables sont choisies dans un ensemble fini de valeurs). Dans notre cas nous utilisons des domaines finis qui sont des intervalles d'entiers. Un solveur PLC fournit un ensemble de contraintes prédéfinies, mais il est aussi possible de créer ses propres contraintes.

Un utilisateur de la technologie PLC sur domaines finis doit suivre les deux étapes suivantes pour formaliser son problème :

1. déclarer les variables du problème et leurs domaines,
2. spécifier un ensemble de contraintes sur ces variables.

Exemple A.1 Supposons que l'utilisateur déclare deux variables X et Y prenant des valeurs entières dans les intervalles $[0, 2]$ et $[0, 1]$. Les six paires de valeurs (X, Y) compatibles avec les intervalles donnés sont $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$, $(2, 0)$ et $(2, 1)$. Elles sont représentées par les points dans la Figure A.1.1(a).

Supposons que l'utilisateur impose les 2 contraintes $X > Y$ et $X + Y \neq 2^4$. L'ensemble de paires de valeurs (X, Y) qui satisfont ces contraintes est alors $\{(1, 0), (2, 1)\}$ (représenté dans la Figure A.1.1(c)).

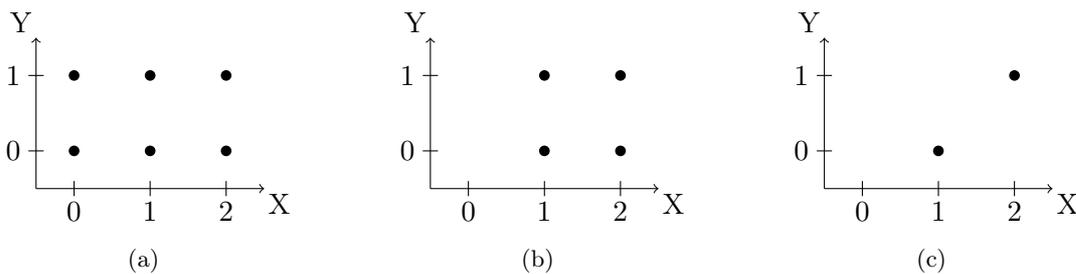


FIG. A.1 – Représentation graphique des solutions potentielles relatives à l'ensemble de contraintes de l'Exemple A.1 (a) après la déclaration des domaines des variables, (b) après propagation des contraintes, et (c) après énumération des solutions instanciées.

La construction de base en PLC est la définition de prédicats à l'aide d'implications universelles sur des formules atomiques portant sur ce prédicat. La déclaration est définie syntaxiquement par :

²⁴Il est à noter qu'il n'est pas nécessaire en PLC que les contraintes soient linéaires comme dans cet exemple simpliste permettant d'illustrer facilement l'ensemble des solutions.

<formule atomique> \Leftarrow <conjonction de contraintes ou de formules atomiques portant sur les paramètres de la formule atomique définie>

Exemple A.2 Nous pouvons définir le prédicat p avec comme paramètres U et V par :

$$p(U, V) \Leftarrow (U > V \wedge U + V \neq 2)$$

Dans un contexte où X et Y sont déclarées dans les intervalles $[0, 2]$ et $[0, 1]$, il est possible d'utiliser ce prédicat de multiples manières pour poser des contraintes : par exemple à l'aide des formules atomiques $p(X, Y)$, $p(X, X)$, $p(1, Y)$ ou encore de la conjonction d'une formule atomique et d'une contrainte $p(X, Y) \wedge X = 1$. L'ensemble de paires de valeurs qui satisfont $p(X, Y)$ est $\{(X, Y) = (1, 0), (X, Y) = (2, 1)\}$ (représenté en Figure A.1.1(c)). L'ensemble de valeurs de X qui satisfont $p(X, X)$ est $\{\}$ (X ne pouvant pas être strictement supérieur à lui même). L'ensemble de valeurs de Y qui satisfont $p(1, Y)$ est $\{Y = 0\}$. L'ensemble de paires de valeurs qui satisfont $p(X, Y) \wedge X = 1$ est $\{(X, Y) = (1, 0)\}$.

A.2 Aspect opérationnel

La PLC permet aux utilisateurs d'obtenir des réponses à leurs requêtes sous la forme d'un ensemble d'intervalles réduits des variables. Nous expliquons dans cette section (i) le sens des réponses du solveur à la partie déclarative de la requête (présentée dans la sous-section A.1), (ii) la technique principale de réduction des intervalles par les solveurs PLC, et enfin (iii) la manière de procéder pour contrôler plus finement le format des solutions de l'interpréteur.

Il est important de clarifier les propriétés suivantes à propos de la réponse à une requête (composée uniquement des deux étapes présentées dans la partie consacrée à l'aspect déclaratif) faite par un solveur PLC sur un ensemble de variables et un ensemble de contraintes sur ces dernières :

1. Si le solveur PLC fournit une réponse négative alors il n'y a pas de solution à la requête.
2. Si le solveur PLC fournit une réponse positive où toutes les variables sont assignées alors cette solution est l'unique solution.
3. Si le solveur PLC fournit une réponse positive où au moins une variable n'est pas assignée (une réponse non instanciée), alors s'il y a une solution à la requête alors les valeurs des variables pour cette solution sont dans les intervalles de valeurs donnés par cette réponse non instanciée.

Les solveurs PLC sur intervalles d'entiers procèdent usuellement par réduction des intervalles des variables de telle sorte que les contraintes soient vérifiées aux bornes de ces intervalles.

Exemple A.3 Dans le contexte où X est dans $[0, 2]$ et Y dans $[0, 1]$ l'imposition de la contrainte $p(X, Y)$ (où p est défini comme dans l'Exemple A.2) permet de réduire l'intervalle de X à $[1, 2]$. Ceci est réalisé par le solveur lorsqu'il contrôle les bornes de X et Y vis-à-vis de la contrainte $X > Y$ dans le contexte où Y est dans l'intervalle $[0, 1]$ car la valeur min de X doit être strictement supérieure à la valeur min de Y . Aucune autre réduction n'est réalisée pour cette requête (une explication simple vient du fait que les deux solutions attendues A.1.1(c) utilisent toutes les valeurs restantes des intervalles de X et Y). Le solveur répond positivement en fournissant les intervalles de valeurs de X et Y ; dans ce cas $X \in [1, 2]$ et $Y \in [0, 1]$. La propriété 3 est alors applicable à ce cas. Les solutions potentielles sont représentées dans la Figure A.1.1(b).

Dans le cas de la propriété 3 l'utilisateur peut définir et utiliser dans sa requête une heuristique d'énumération des valeurs des variables afin d'obtenir les solutions instanciées.

Exemple A.4 *Considérons l'Exemple A.3. Supposons que l'utilisateur souhaite déterminer toutes les solutions complètement instanciées. Il indique alors par exemple au solveur d'énumérer toutes les valeurs de toutes les variables en commençant par les valeurs de X puis celles de Y dans l'ordre croissant.*

Dans ce cas, le solveur commence par instancier X et Y à 1 et 0 respectivement et trouve alors une cohérence. Puis il revient en arrière et instancie X à 1 et Y à 1. Le solveur trouve alors une incohérence en particulier grâce à la contrainte $X > Y$. Puis le solveur trouve une cohérence pour $X = 2$ et $Y = 0$. Enfin, le solveur trouve une incohérence pour $X = 2$ et $Y = 1$ grâce à la contrainte $X + Y \neq 2$. Finalement la résolution se termine étant donné que tous les couples de valeurs dans les intervalles de X et Y ont été testés. Ainsi l'utilisateur obtient les solutions $(X, Y) = (1, 0)$ et $(X, Y) = (2, 1)$, représentées dans la Figure A.1.1(c).

Annexe B

Description continue et description discrète selon R. Thomas de réseaux génétiques

Considérons un graphe \mathcal{G} d'interactions avec n gènes c_i , $i \in [1, n]$. L'évolution continue de la concentration c_i avec le temps est exprimée usuellement par une équation différentielle du type suivant :

$$\dot{c}_i = \left(\sum_k \kappa_{c_i, k} * b_{i, k}(\bar{c}) \right) - \lambda_{c_i} * c_i \quad (\text{B.1})$$

où \bar{c} est le vecteur des variables de concentration c_i , $\kappa_{c_i, k}$ un coefficient positif représentant le taux de production du produit du i^{eme} gène, et λ_{c_i} un coefficient positif représentant le taux de dégradation du produit du i^{eme} gène. De plus, chaque $b_{i, k}(\bar{c})$ est une fonction composée de sommes de produits de sigmoïdes portant sur les concentrations des produits des gènes interagissant sur le i^{eme} gène. Une sigmoïde positive, respectivement négative, de point d'inflexion $c_j = t_{c_j}^p$ est représentée par $s^+(c_j, t_{c_j}^p)$, respectivement par $s^-(c_j, t_{c_j}^p)$. Une fonction $b_{i, k}(\bar{c})$ ne peut faire intervenir une sigmoïde $s^\sigma(c_j, t_{c_j}^p)$, $\sigma \in \{-, +\}$, sue s'il existe dans le graphe \mathcal{G} une arête $(c_i, t_{c_j}^p, c_j)$.

Exemple B.5 *Pour l'exemple de graphe \mathcal{G} d'interactions de la Figure 3.1 le système d'équations différentielles \mathcal{E} peut être le suivant :*

$$\mathcal{E} = \begin{cases} \dot{x} &= \kappa_{x,1} * s^-(y, t_y^1) - \lambda_x * x \\ \dot{y} &= \kappa_{y,1} * s^+(x, t_x^1) + \kappa_{y,2} * s^+(y, t_y^2) - \lambda_y * y \end{cases} \quad (\text{B.2})$$

Un système d'équations différentielles de ce type, noté \mathcal{E} , précise donc \mathcal{G} en introduisant des paramètres relatifs aux taux de production et dégradation mais aussi en spécifiant un ensemble de compositions additives d'interactions (voir l'Exemple B.9 en fin de cette Annexe). \mathcal{E} est alors moins général qu'un modèle discret de Thomas \mathcal{M} .

La discrétisation d'un système d'équations \mathcal{E} menant à un modèle discret \mathcal{M} comporte trois étapes : la première définit \mathcal{E}' comme le système d'équations différentielles linéaire par morceaux où chaque sigmoïde de \mathcal{E} est discrétisée en une fonction Booléenne, la deuxième définit \mathcal{E}'' comme le système d'équations constantes par morceaux établissant la tendance du système dans un état \bar{c} , la troisième définit le modèle \mathcal{M} associé à \mathcal{E}'' .

La première étape de discrétisation est un remplacement de chaque sigmoïde de \mathcal{E} par une fonction Booléenne dans \mathcal{E}' , \mathcal{E} et \mathcal{E}' étant pour le reste identiques. \mathcal{E}' est du type :

$$\dot{c}_i = \left(\sum_k \kappa_{c_i,k} * b'_{i,k}(\bar{c}) \right) - \lambda_{c_i} * c_i \quad (\text{B.3})$$

où chaque $b'_{i,k}(\bar{c})$ est la fonction déduite de $b_{i,k}(\bar{c})$ en remplaçant chaque sigmoïde $s^+(c_j, t_{c_j}^p)$ par $c_j \geq t_{c_j}^p$ et chaque sigmoïde $s^-(c_j, t_{c_j}^p)$ par $c_j < t_{c_j}^p$.

Exemple B.6 Pour le système d'équations différentielles \mathcal{E} de l'Exemple B.5 nous obtenons :

$$\mathcal{E}' = \begin{cases} \dot{x} &= \kappa_{x,1} * (y < t_y^1) - \lambda_x * x \\ \dot{y} &= \kappa_{y,1} * (x \geq t_x^1) + \kappa_{y,2} * (y \geq t_y^2) - \lambda_y * y \end{cases} \quad (\text{B.4})$$

La deuxième étape introduit la notion de tendance. La tendance de c_i dans un état \bar{c} est donné par la solution en c_i de l'équation $\dot{c}_i = 0$. Cette tendance est notée f_{c_i} . Le système d'équations obtenu est nommé \mathcal{E}'' . Si le système se trouve dans l'état \bar{c} il évolue vers l'état $\langle f_{c_1}, \dots, f_{c_n} \rangle$, dits état focal de \bar{c} .

Exemple B.7 Pour le système d'équations \mathcal{E}' nous obtenons le système d'équations \mathcal{E}'' suivant définissant $\langle f_x, f_y \rangle$:

$$\mathcal{E}'' = \begin{cases} f_x &= (\kappa_{x,1}/\lambda_x) * (y < t_y^1) \\ f_y &= (\kappa_{y,1}/\lambda_y) * (x \geq t_x^1) + (\kappa_{y,2}/\lambda_y) * (y \geq t_y^2) \end{cases} \quad (\text{B.5})$$

La troisième étape consiste à discrétiser dans \mathcal{E}'' toutes les variables c_i , f_{c_i} , $t_{c_i}^p$ et somme de quotient $\kappa_{c_i,k}/\lambda_{c_i}$ en un nombre fini de valeurs entières positives. Les somme de quotients $\kappa_{c_i,k}/\lambda_{c_i}$ sont notés $K_{c_i}^l$. Toutes les valeurs entières c_i , f_{c_i} et $K_{c_i}^l$ sont définies dans des intervalles d'entiers $[0, \max_{c_i}]$ et $t_{c_i}^p = p, p \in [1, \max_{c_i}]$, où \max_{c_i} est la valeur maximale de l'indice p des $t_{c_i}^p$. Nous obtenons un système d'équations numériques \mathcal{E}''' définissant un état focal pour tout état, où un état est un hyper-cube de l'espace des concentrations délimité par les seuils $t_{c_i}^p$.

Exemple B.8 Pour le système d'équations \mathcal{E}'' nous obtenons le système d'équations numérique \mathcal{E}''' suivant :

$$\mathcal{E}''' = \begin{cases} f_x &= K_x^1 * (y \geq t_y^1) + \\ &K_x^2 * (y < t_y^1) \\ f_y &= K_y^1 * (x < t_x^1) * (y < t_y^2) + \\ &K_y^2 * (x < t_x^1) * (y \geq t_y^2) + \\ &K_y^3 * (x \geq t_x^1) * (y < t_y^2) + \\ &K_y^4 * (x \geq t_x^1) * (y \geq t_y^2) \end{cases} \quad (\text{B.6})$$

où K_x^2 est la discrétisation de $\kappa_{x,1}/\lambda_x$, K_y^2 celle de $\kappa_{y,2}/\lambda_y$, K_y^3 celle de $\kappa_{y,1}/\lambda_y$, et K_y^4 celle de $(\kappa_{y,1}/\lambda_y) + (\kappa_{y,2}/\lambda_y)$.

Le système d'équation \mathcal{E} implique des hypothèses sur les compositions d'interactions. De plus ces compositions d'interactions sont systématiquement additives.

Exemple B.9 Dans l'Exemple B.5 l'équation définissant \dot{x} contient seulement l'expression $(y < t_y^1)$. Celle-ci indique que la seule composition d'interactions supposée sur x est une inhibition par y à partir de son seuil t_y^1 . De la même manière, concernant y , \mathcal{E} indique deux compositions d'interactions : “ x active y à partir de son seuil t_x^1 ” et “ y s'active lui-même à partir de son seuil t_y^2 ”.

Ces hypothèses de compositions d'interactions entraîne les contraintes d'additivité suivantes : $K_x^1 \leq K_x^2$, $K_y^1 \leq K_y^2$, $K_y^1 \leq K_y^3$, $K_y^2 \leq K_y^4$, $K_y^3 \leq K_y^4$. Il est à noter que ce sont les mêmes contraintes que celles de l'Exemple 4.8.

De plus, le fait que la tendance de la concentration des produits des gènes x et y est nulle dans les états où les compositions d'interactions relatives à ces produits sont inactives entraîne les contraintes $K_x^1 = 0$ et $K_y^1 = 0$.

Finalement, le modèle \mathcal{M} est le couple composé de \mathcal{E}''' et de l'ensemble des contraintes d'additivité et des contraintes spécifiant que les tendances “basales” sont égales à 0.

Cette discrétisation a été proposée par R. Thomas [52].

Annexe C

Démonstration de la Contrainte 4.3

Nous proposons ici une démonstration de l'équivalence entre la Définition 4.6 de l'observabilité d'une composition d'interactions en terme de \mathcal{T} et la propriété en terme de \mathcal{P} donné par la Contrainte 4.3. Pour montrer cette équivalence nous procédons en deux étapes en démontrant chacune des deux implications.

C.1 La Contrainte 4.3 est nécessaire

Soit I une composition d'interaction observable suivant la Définition 4.6 pour le couple d'états $(S0, S1)$. $S0$ et $S1$ sont alors adjacents, et I est inactive dans $S0$ tandis qu'elle est active dans $S1$.

Trois cas se présentent suivant la Définition 4.6 :

1. Soit $S0$ et $S1$ différent sur une composante différente de c . Nous avons donc $C_{c,S0} = C_{c,S1}$. Deux sous-cas se présentent :
 - il existe une transition partant de $S1$ augmentant suivant c et il n'existe pas de transition partant de $S0$ augmentant suivant c .
 Pour $S = S1$ et S' adjacent à S tel que $C_{c,S'} = C_{c,S} + 1$ nous obtenons la contrainte $transition_rule(t(S, S'), F_S) \Leftrightarrow (C_{c,S} < F_{c,S})$ en simplifiant la Contrainte 3.3 dans le contexte où $LBup_c$ est vrai. Comme il doit exister une transition de S vers S' alors $transition(M, t(S, S'), P)$ est vrai donc $transition_rule(t(S, S'), F_S)$ et donc $C_{c,S} < F_{c,S}$. Finalement il existe $l1$ tel que $Cellc_{c,S1}^{l1}$ est vrai, donc $C_{c,S1} < K_c^{l1} = F_{c,S1}$.
 De la même manière, pour $S = S0$ et S' adjacent à S tel que $C_{c,S'} = C_{c,S} + 1$ nous obtenons la contrainte $transition_rule(t(S, S'), F_S) \Leftrightarrow (C_{c,S} < F_{c,S})$. Comme il ne doit pas exister de transition de S vers S' alors $transition_rule(t(S, S'), F_S)$ est faux et donc $F_{c,S} \leq C_{c,S}$. Finalement il existe $l0$ tel que $Cellc_{c,S0}^{l0}$ est vrai, donc $K_c^{l0} = F_{c,S0} \leq C_{c,S0}$. Etant donné que $C_{c,S0} = C_{c,S1}$ nous obtenons la propriété $K_c^{l0} \leq C_{c,S0} = C_{c,S1} < K_c^{l1}$ qui implique $K_c^{l0} < K_c^{l1}$.
 - il existe une transition partant de $S0$ diminuant suivant c et il n'existe pas de transition partant de $S1$ diminuant suivant c .
 De manière symétrique au cas précédent nous pouvons montrer l'existence d'un couple de conditions $(Cellc_{c,S}^{l0}, Cellc_{c,S}^{l1})$ tel que $Cellc_{c,S}^{l0}$ et $Cellc_{c,S}^{l1}$ sont vraies, et tel que les propriétés $C_{c,S1} \leq K_c^{l1} = F_{c,S1}$ et $K_c^{l0} = F_{c,S0} < C_{c,S0}$ sont vraies.
 Etant donné que $C_{c,S0} = C_{c,S1}$ nous obtenons la propriété $K_c^{l0} < C_{c,S0} = C_{c,S1} \leq K_c^{l1}$ incluant la propriété $K_c^{l0} < K_c^{l1}$.

2. Soit $S0$ et $S1$ différent sur la composante c tel que $C_{c,S0} < C_{c,S1}$.

Pour $S = S1$ et S' adjacent à S tel que $C_{c,S'} = C_{c,S} - 1$ nous obtenons la contrainte $transition_rule(t(S, S'), F_S) \Leftrightarrow (F_{c,S} < C_{c,S})$. Comme il ne doit pas exister de transition de S vers S' alors $transition(M, t(S, S'), P)$ est faux, et donc $\neg transition_rule(t(S, S'), F_S)$, et donc $C_{c,S} \leq F_{c,S}$. Finalement il existe $l1$ tel que $Cellc_{c,S1}^{l1}$ est vrai, et donc $C_{c,S1} \leq F_{c,S1} = K_c^{l1}$.

De la même manière, pour $S = S0$ et S' adjacent à S tel que $C_{c,S'} = C_{c,S} + 1$ nous obtenons la contrainte $transition_rule(t(S, S'), F_S) \Leftrightarrow (C_{c,S} < F_{c,S})$. Comme il ne doit pas exister de transition de S vers S' alors $transition_rule(t(S, S'), F_S)$ est faux pour F_S l'état focal de S et donc $F_{c,S} \leq C_{c,S}$. Finalement il existe $l0$ tel que $Cellc_{c,S0}^{l0}$ est vrai, et donc $K_c^{l0} = F_{c,S0} \leq C_{c,S0}$.

Etant donné que $C_{c,S0} < C_{c,S1}$ nous obtenons la propriété $K_c^{l0} \leq C_{c,S0} < C_{c,S1} \leq K_c^{l1}$ qui implique $K_c^{l0} < K_c^{l1}$.

3. Soit $S0$ et $S1$ différent sur la composante c et tels que $C_{c,S0} > C_{c,S1}$.

De manière symétrique au cas précédent nous pouvons montrer l'existence d'un couple de conditions $(Cellc_{c,S}^{l0}, Cellc_{c,S}^{l1})$ tel que $Cellc_{c,S0}^{l0}$ et $Cellc_{c,S1}^{l1}$ sont vraies et tel que les propriétés $C_{c,S0} \leq F_{c,S1} = K_c^{l1}$ et $K_c^{l0} = F_{c,S0} \leq C_{c,S1}$ sont vraies.

Etant donné que $C_{c,S0} > C_{c,S1}$ nous obtenons la propriété $K_c^{l0} \leq C_{c,S1} < C_{c,S0} \leq K_c^{l1}$ qui implique $K_c^{l0} < K_c^{l1}$.

Ainsi pour tous les cas de la Définition 4.6 il existe un couple $(Cellc_{c,S}^{l0}, Cellc_{c,S}^{l1})$ tel que $Cellc_{c,S0}^{l0}$ et $Cellc_{c,S1}^{l1}$ sont vraies et tel que $K_c^{l0} < K_c^{l1}$ est vraie.

Par définition de $min_{c,l}$ et $max_{c,l}$ (voir Section 4.1.3) nous avons $C_{c,S0} \leq max_{c,l0}$, $C_{c,S1} \leq max_{c,l1}$, $min_{c,l0} \leq C_{c,S0}$ et $min_{c,l1} \leq C_{c,S1}$. Pour tous les cas précédemment vus nous avons la propriété $K_c^{l0} \leq C_{c,S0}$. Donc $K_c^{l0} \leq max_{c,l0}$ est vraie. De la même manière pour tous les cas précédemment vus nous avons $K_c^{l0} \leq C_{c,S1}$, $C_{c,S0} \leq K_c^{l1}$ et $C_{c,S1} \leq K_c^{l1}$. Donc $K_c^{l0} \leq max_{c,l1}$, $min_{c,l0} \leq K_c^{l1}$ et $min_{c,l1} \leq K_c^{l1}$ sont vraies.

Enfin $Cellc_{c,S0}^{l0}$ et $Cellc_{c,S1}^{l1}$ sont vraies. $S0$ et $S1$ étant adjacents, $Cellc_{c,S}^{l0}$ et $Cellc_{c,S}^{l1}$ sont adjacents. La composition d'interactions I est inactive dans $S0$ et donc inactive dans $Cellc_{c,S}^{l0}$. La composition d'interactions I est active dans $S1$ et donc active dans $Cellc_{c,S}^{l1}$.

En conclusion, la définition 4.6 implique bien l'existence d'un couple $(Cellc_{c,S}^{l0}, Cellc_{c,S}^{l1})$ vérifiant toutes les propriétés énoncées dans la Contrainte 4.3.

C.2 La Contrainte 4.3 est suffisante

Supposons que I est une composition d'interaction satisfaisant la Contrainte 4.3 pour le couple $(Cellc_{c,S}^{l0}, Cellc_{c,S}^{l1})$. Les contextes cellulaires identifié par $Cellc_{c,S}^{l0}$ et $Cellc_{c,S}^{l1}$ sont alors adjacents, et I est inactive dans $Cellc_{c,S}^{l0}$ tandis qu'elle est active dans $Cellc_{c,S}^{l1}$.

Considérons alors trois cas suivants analogues aux trois cas de la Définition 4.6 en considérant les positions respectives des contextes cellulaires identifiés par $Cellc_{c,S}^{l0}$ et $Cellc_{c,S}^{l1}$ (ces contextes cellulaires n'étant pas vides car ils sont adjacents) :

1. Soient $Cellc_{c,S}^{l0}$ et $Cellc_{c,S}^{l1}$ définissant deux régions rectangulaires séparées par un seuil t sur une composante c' différente de c . Nous avons alors les propriétés $min_{c,l0} = min_{c,l1}$ et $max_{c,l0} = max_{c,l1}$. Décomposons alors ce cas en trois sous-cas suivant un découpage des valeurs possibles de K_c^{l0} et K_c^{l1} étant donné les propriétés de la Contrainte 4.3 :

- Soit $K_c^{l1} > \max_{c,l1}$. Choisissons les états $S0$ et $S1$ adjacents séparés par le seuil t tels que $C_{c,S0} = C_{c,S1} = \max_{c,l1}$ et tels que $Cellc_{c,S0}^{l0}$ et $Cellc_{c,S1}^{l1}$ sont vraies. Nous avons alors la propriété $C_{c,S1} = \max_{c,l1} < K_c^{l1} = F_{c,S1}$. Etant donné que $K_c^{l0} \leq \max_{c,l1}$ nous avons aussi la propriété $F_{c,S0} = K_c^{l0} \leq \max_{c,l1} = C_{c,S0}$.
Pour $S = S1$ et S' adjacent à S tel que $C_{c,S'} = C_{c,S} + 1$ nous obtenons la contrainte $transition_rule(t(S, S'), F_S) \Leftrightarrow (C_{c,S} < F_{c,S})$ en simplifiant la Contrainte 3.3 dans le contexte où $LBup_c$ est vrai. Sachant $C_{c,S1} < F_{c,S1}$ nous obtenons la propriété $transition_rule(t(S, S'), F_S)$ indiquant qu'il existe une transition partant de $S1$ augmentant suivant c .
De la même manière pour $S = S0$ et S' adjacent à S tel que $C_{c,S'} = C_{c,S} + 1$ nous obtenons la contrainte $transition_rule(t(S, S'), F_S) \Leftrightarrow C_{c,S} < F_{c,S}$. Sachant $F_{c,S0} \leq C_{c,S0}$ nous obtenons la propriété $\neg transition_rule(t(S, S'), F_S)$ indiquant qu'il n'existe pas de transition partant de $S0$ augmentant suivant c .
Finalement nous avons donc trouvé $S0$ et $S1$ adjacents tels que I est inactive dans $S0$ et I est active dans $S1$ et satisfaisant la condition du premier sous-cas du premier cas de la Définition 4.6.
- Soit $\min_{c,l0} > K_{c,l0}$. Choisissons les états $S0$ et $S1$ adjacents séparés par le seuil t tels que $C_{c,S0} = C_{c,S1} = \min_{c,l0}$ et tels que $Cellc_{c,S0}^{l0}$ et $Cellc_{c,S1}^{l1}$ sont vraies. Nous avons alors la propriété $F_{c,S0} = K_c^{l0} < \min_{c,l0} = C_{c,S0}$. Etant donné que $\min_{c,l0} \leq K_c^{l1}$ nous avons aussi la propriété $F_{c,S1} = K_c^{l1} \geq \min_{c,l0} = C_{c,S0}$.
De la même manière que le sous-cas précédent, pour $S = S0$ et S' adjacent à S tel que $C_{c,S'} = C_{c,S} - 1$ nous obtenons la propriété $transition_rule(t(S, S'), F_S)$ indiquant qu'il existe une transition partant de $S0$ diminuant suivant c .
Pour $S = S1$ et S' adjacent à S tel que $C_{c,S'} = C_{c,S} - 1$ nous obtenons la propriété $\neg transition_rule(t(S, S'), F_S)$ indiquant qu'il n'existe pas de transition partant de $S0$ diminuant suivant c .
Finalement nous avons donc trouvé $S0$ et $S1$ adjacents et tels que I est inactive dans $S0$ et I est active dans $S1$ et satisfaisant la condition du deuxième sous-cas du premier cas de la Définition 4.6.
- Soit $\min_{c,l0} \leq K_c^{l0} < K_c^{l1} \leq \max_{c,l1}$. Soient les états $S0$ et $S1$ adjacents séparés par le seuil t tels que $C_{c,S0} = C_{c,S1} = K_{c,S0}$ et tels que $Cellc_{c,S0}^{l0}$ et $Cellc_{c,S1}^{l1}$ sont vrais. Nous avons alors les propriétés $F_{c,S0} = K_c^{l0} = C_{c,S0}$ et $C_{c,S1} = K_c^{l0} < K_c^{l1} = F_{c,S1}$.
Pour $S = S1$ et S' adjacent à S tel que $C_{c,S'} = C_{c,S} + 1$ nous obtenons la contrainte $transition_rule(t(S, S'), F_S) \Leftrightarrow C_{c,S} < F_{c,S}$. Sachant $C_{c,S1} < F_{c,S1}$ nous obtenons la propriété $transition_rule(t(S, S'), F_S)$ indiquant qu'il existe une transition partant de $S1$ augmentant suivant c .
Pour $S = S0$ et S' adjacent à S tel que $C_{c,S'} = C_{c,S} + 1$ nous obtenons la contrainte $transition_rule(t(S, S'), F_S) \Leftrightarrow C_{c,S} < F_{c,S}$. Sachant $F_{c,S0} = C_{c,S0}$ nous obtenons la propriété $\neg transition_rule(t(S, S'), F_S)$ indiquant qu'il n'existe pas de transition partant de $S0$ augmentant suivant c .
Finalement nous avons donc trouvé $S0$ et $S1$ adjacents tels que I est inactive dans $S0$ et I est active dans $S1$ et satisfaisant la condition du premier sous-cas du premier cas de la Définition 4.6²⁵.

2. Soit $Cellc_{c,S}^{l0}$ et $Cellc_{c,S}^{l1}$ définissant deux régions rectangulaires séparées par un seuil t sur la composante c , et telles que la région définie par $Cellc_{c,S}^{l0}$ est dessous celle définie par

²⁵Nous pouvons montrer que le deuxième sous-cas du premier cas de la Définition 4.6 est aussi satisfait en prenant $S0$ et $S1$ tels que $C_{c,S0} = C_{c,S1} = K_{c,S1}$.

$Cellc_{c,S}^{l1}$ suivant c , c'est-à-dire $min_{c,l1} = max_{c,l0} + 1$.

Soient les états $S0$ et $S1$ adjacents séparés par le seuil t tels que $C_{c,S0} = max_{c,l0}$, $C_{c,S1} = min_{c,l1}$ et tels que $Cellc_{c,S0}^{l0}$ et $Cellc_{c,S1}^{l1}$ sont vraies. Nous avons alors les propriétés $F_{c,S0} = K_c^{l0} \leq max_{c,l0} = C_{c,S0}$ et $C_{c,S1} = min_{c,l1} \leq K_c^{l1} = F_{c,S1}$.

Pour $S = S0$ et S' adjacent à S tel que $C_{c,S'} = C_{c,S} + 1$ nous obtenons la contrainte $transition_rule(t(S, S'), F_S) \Leftrightarrow (C_{c,S} < F_{c,S})$. Sachant $F_{c,S0} \leq C_{c,S0}$ nous obtenons la propriété $\neg transition_rule(t(S, S'), F_S)$ indiquant qu'il n'existe pas de transition partant de $S0$ augmentant suivant c .

Pour $S = S1$ et S' adjacent à S tel que $C_{c,S'} = C_{c,S} - 1$ nous obtenons la contrainte $transition_rule(t(S, S'), F_S) \Leftrightarrow (F_{c,S} < C_{c,S})$. Sachant $C_{c,S1} \leq F_{c,S1}$ nous obtenons la propriété $\neg transition_rule(t(S, S'), F_S)$ indiquant qu'il n'existe pas de transition partant de $S1$ diminuant suivant c .

Finalement il existe $S0$ et $S1$ adjacents tels que I est inactive dans $S0$ et I est active dans $S1$ et satisfaisant la condition du deuxième cas de la Définition 4.6.

3. Soit $Cellc_{c,S}^{l0}$ et $Cellc_{c,S}^{l1}$ définissant deux régions rectangulaires séparées par un seuil t sur la composante c , et telles que la région définie par $Cellc_{c,S}^{l0}$ est dessus celle définie par $Cellc_{c,S}^{l1}$ suivant c , c'est-à-dire $min_{c,l0} = max_{c,l1} + 1$.

Soient les états $S0$ et $S1$ adjacents séparés par le seuil t tels que $C_{c,S0} = min_{c,l0}$, $C_{c,S1} = max_{c,l1}$ et tels que $Cellc_{c,S0}^{l0}$ et $Cellc_{c,S1}^{l1}$ sont vraies. Nous avons alors les propriétés $F_{c,S0} = K_c^{l0} \leq max_{c,l1} = C_{c,S1} < C_{c,S0}$ et $C_{c,S1} < C_{c,S0} = min_{c,l0} \leq K_c^{l1} = F_{c,S1}$.

Pour $S = S0$ et S' adjacent à S tel que $C_{c,S'} = C_{c,S} - 1$ nous obtenons la contrainte $transition_rule(t(S, S'), F_S) \Leftrightarrow (F_{c,S} < C_{c,S})$. Sachant $F_{c,S0} < C_{c,S0}$ nous obtenons la propriété $transition_rule(t(S, S'), F_S)$ indiquant qu'il existe une transition partant de $S0$ diminuant suivant c .

Pour $S = S1$ et S' adjacent à S tel que $C_{c,S'} = C_{c,S} + 1$ nous obtenons la contrainte $transition_rule(t(S, S'), F_S) \Leftrightarrow (C_{c,S} < F_{c,S})$. Sachant $C_{c,S1} < F_{c,S1}$ nous obtenons la propriété $transition_rule(t(S, S'), F_S)$ indiquant qu'il existe une transition partant de $S1$ augmentant suivant c .

Finalement il existe $S0$ et $S1$ adjacents tels que I est inactive dans $S0$ et I est active dans $S1$ et satisfaisant la condition du troisième cas de la Définition 4.6.

En conclusion, si la Contrainte 4.3 est vraie il existe un couple d'états $(S0, S1)$ vérifiant toutes les propriétés de la Définition 4.6.

Annexe D

Fichiers Prolog du graphe d'interaction et du modèle de l'exemple de la Table 3.1

D.1 Fichier Prolog du graphe d'interaction \mathcal{G} de la Table 3.1

```
/*
*/
:- module(premodel_chaos, [premodel/1]).

premodel(Premodel) :-
    Premodel = [Species, LI, Nb_thr, Thr_ord],
    Species = [x,y],

    LI =
    [
        c(x,[i(y,1)]),
        c(y,[i(x,1),i(y,2)])
    ],

    Nb_thr = [1,2],
    Thr_ord = [thr_ord(y,[1,2])].
```

10

D.2 Fichier Prolog du modèle \mathcal{M} de la Table 3.1

```
/*
*/
:- module(model_chaos, [biomodel/6]).
:- use_module(library(clpfd)).
:- use_module(cnf).
:- use_module(biomodel_tool).
:- use_module(library(lists)).
:- use_module(outils).

biomodel(Model, Mutant, Modif_nb_thr, Comp, Relax, Cnf) :-
```

10

```

Model = [Species, Eqs, P, Nb_thr],

Species = [x,y],

Eqs = [
  eq(x,
    [[IKx_1,Kx_1], [IKx_2,Kx_2]],
    [i(y,[ty_1,Ty_1])],
    [[0],[1]]),
  eq(y,
    [[IKy_1,Ky_1], [IKy_2,Ky_2], [IKy_3,Ky_3], [IKy_4,Ky_4]],
    [i(x,[tx_1,Tx_1]), i(y,[ty_2,Ty_2])],
    [[0,0],[0,1],[1,0],[1,1]])
],

Nb_thr = [[nb_thr_x,_Nb_thr_x], [nb_thr_y,_Nb_thr_y]],
Thr = [[[tx_1,Tx_1]], [[ty_1,Ty_1],[ty_2,Ty_2]]],
biomodel_tool :interval_Nb_thr_and_Thr(Species, Nb_thr, Thr, [1,2], Modif_nb_thr, Relax, Cnf1),

P = [ [],
  [[[tx_1,Tx_1]], [[ty_1,Ty_1],[ty_2,Ty_2]]],
  [[[IKx_1,Kx_1], [IKx_2,Kx_2]],[[IKy_1,Ky_1], [IKy_2,Ky_2], [IKy_3,Ky_3], [IKy_4,Ky_4]] ]],
biomodel_tool :names_IK(Species, P, Mutant),

cnf :add_list_int_to_cnf([IKx_1, IKx_2], 0, 1, Cnf2),

cnf :add_list_int_to_cnf([IKy_1, IKy_2, IKy_3, IKy_4], 0, 2, Cnf3),

Thr_ord = [thr_ord(y,[1,2])],
biomodel_tool :ineg_ord_thr(Thr, Species, Thr_ord, Cnf4),

Bnb_thr = [[[_IBnb_thr_x_1,_Bnb_thr_x_1]],[[_IBnb_thr_y_1,_Bnb_thr_y_1], [_IBnb_thr_y_2,_Bnb_thr_y_2]]],
biomodel_tool :constraints_Bnb_thr(Species, Thr, Modif_nb_thr, Relax, Bnb_thr, Nb_thr, Cnf5),

biomodel_tool :constraints_Mutant(Mutant, Eqs, Cnf6),
biomodel_tool :constraints_Relax(Relax, Modif_nb_thr, Bnb_thr, Thr, Species, Cnf7),
biomodel_tool :constraints_Comp(Comp, Relax, Mutant, Eqs, Cnf8),

Kx_1 in 0 .. 1,
Kx_2 in 0 .. 1,
Ky_1 in 0 .. 2,
Ky_2 in 0 .. 2,
Ky_3 in 0 .. 2,
Ky_4 in 0 .. 2,

biomodel_tool :constraints_P_K_Nb_thr(P, Nb_thr, Cnf9),
biomodel_tool :constraints_P_Input_Nb_thr(P, Nb_thr, Cnf10),

append([Cnf1,Cnf2,Cnf3,Cnf4,Cnf5,Cnf6,Cnf7,Cnf8,Cnf9,Cnf10], Cnf).

```

Annexe E

Equations focales des modèles des deux premières applications du Chapitre 6

E.1 Equations focales du modèle de contrôle de l'immunité par le phage λ

$$\begin{aligned}
 F_{cI,S} = & K_{cI}^1 \cdot S_{cI} < t_{cI}^2 \wedge S_{cro} < t_{cro}^1 \wedge S_{cII} < t_{cII}^1 + \\
 & K_{cI}^2 \cdot S_{cI} < t_{cI}^2 \wedge S_{cro} < t_{cro}^1 \wedge S_{cII} \geq t_{cII}^1 + \\
 & K_{cI}^3 \cdot S_{cI} < t_{cI}^2 \wedge S_{cro} \geq t_{cro}^1 \wedge S_{cII} < t_{cII}^1 + \\
 & K_{cI}^4 \cdot S_{cI} < t_{cI}^2 \wedge S_{cro} \geq t_{cro}^1 \wedge S_{cII} \geq t_{cII}^1 + \\
 & K_{cI}^5 \cdot S_{cI} \geq t_{cI}^2 \wedge S_{cro} < t_{cro}^1 \wedge S_{cII} < t_{cII}^1 + \\
 & K_{cI}^6 \cdot S_{cI} \geq t_{cI}^2 \wedge S_{cro} < t_{cro}^1 \wedge S_{cII} \geq t_{cII}^1 + \\
 & K_{cI}^7 \cdot S_{cI} \geq t_{cI}^2 \wedge S_{cro} \geq t_{cro}^1 \wedge S_{cII} < t_{cII}^1 + \\
 & K_{cI}^8 \cdot S_{cI} \geq t_{cI}^2 \wedge S_{cro} \geq t_{cro}^1 \wedge S_{cII} \geq t_{cII}^1
 \end{aligned}$$

$$\begin{aligned}
 F_{cro,S} = & K_{cro}^1 \cdot S_{cI} < t_{cI}^2 \wedge S_{cro} < t_{cro}^3 + \\
 & K_{cro}^2 \cdot S_{cI} < t_{cI}^2 \wedge S_{cro} \geq t_{cro}^3 + \\
 & K_{cro}^3 \cdot S_{cI} \geq t_{cI}^2 \wedge S_{cro} < t_{cro}^3 + \\
 & K_{cro}^4 \cdot S_{cI} \geq t_{cI}^2 \wedge S_{cro} \geq t_{cro}^3
 \end{aligned}$$

$$\begin{aligned}
 F_{cII,S} = & K_{cII}^1 \cdot S_{cI} < t_{cI}^2 \wedge S_{cro} < t_{cro}^3 \wedge S_n < t_n^1 + \\
 & K_{cII}^2 \cdot S_{cI} < t_{cI}^2 \wedge S_{cro} < t_{cro}^3 \wedge S_n \geq t_n^1 + \\
 & K_{cII}^3 \cdot S_{cI} < t_{cI}^2 \wedge S_{cro} \geq t_{cro}^3 \wedge S_n < t_n^1 + \\
 & K_{cII}^4 \cdot S_{cI} < t_{cI}^2 \wedge S_{cro} \geq t_{cro}^3 \wedge S_n \geq t_n^1 + \\
 & K_{cII}^5 \cdot S_{cI} \geq t_{cI}^2 \wedge S_{cro} < t_{cro}^3 \wedge S_n < t_n^1 + \\
 & K_{cII}^6 \cdot S_{cI} \geq t_{cI}^2 \wedge S_{cro} < t_{cro}^3 \wedge S_n \geq t_n^1 + \\
 & K_{cII}^7 \cdot S_{cI} \geq t_{cI}^2 \wedge S_{cro} \geq t_{cro}^3 \wedge S_n < t_n^1 + \\
 & K_{cII}^8 \cdot S_{cI} \geq t_{cI}^2 \wedge S_{cro} \geq t_{cro}^3 \wedge S_n \geq t_n^1
 \end{aligned}$$

$$\begin{aligned}
 F_{n,S} = & K_n^1 \cdot S_{cI} < t_{cI}^1 \wedge S_{cro} < t_{cro}^2 + \\
 & K_n^2 \cdot S_{cI} < t_{cI}^1 \wedge S_{cro} \geq t_{cro}^2 + \\
 & K_n^3 \cdot S_{cI} \geq t_{cI}^1 \wedge S_{cro} < t_{cro}^2 + \\
 & K_n^4 \cdot S_{cI} \geq t_{cI}^1 \wedge S_{cro} \geq t_{cro}^2
 \end{aligned}$$

Annexe F

Fichiers Prolog des modules *dynamic_model* et *focal_state*

F.1 Fichier Prolog du module *dynamic_model*

```
:-module(dynamic_model, [path/3, transition/3]).

:- use_module(library(clpfd)).
:- use_module(library(lists)).

:- use_module(cnf).
:- use_module(sp_minisat).
:- use_module(focal_state, [focal_state/6]).

:- use_module(constraints_CP_SAT).
:- use_module(interface_CP_SAT).
:- use_module(enumeration).

/*****
dynamic_model(Model, Path_aux, Cnf) true if
* Path_aux is a possible path of states for the biological model Model (C)
* Cnf is the list of clauses equivalent to the satisfiability of C
*/
path(Model, [State|Path], Cnf) :-
    Model = [Species, Eqs, [P_Input,_,_], Nb_thr|_],
    %state_dom(State, Nb_thr, Cnf1),
    path([State|Path], Eqs, Species, P_Input, Nb_thr, Cnf).

path([], -, -, -, -, []).
path([State_i,State_s|Path], Eqs, Species, P_Input, Nb_thr, Cnf) :-
    transition(tr(State_i, State_s), Eqs, Species, P_Input, Nb_thr, Cnf1),
    path([State_s|Path], Eqs, Species, P_Input, Nb_thr, Cnf2),
    append([Cnf1, Cnf2], Cnf).

/*****
transition(State_i, State_s, Eqs, P_input, Nb_thr, [IB,B], Cnf) is true if
* B is equivalent to State_s is a successor of State_i according to :
- the focal equations Eqs
- the input parameters P_input
- the number of thresholds for each species Nb_thr
of the biological model
* Cnf is the list of clauses equivalent to the satisfiability of

```

```

    the previous statement with
    IB the identifier of the variable B in the boolean formula Cnf
*/
transition(tr(State_i, State_s), Eqs, Species, P_Input, Nb_thr, Cnf) :-
    length(Eqs, Nb_no_input),
    lists :append_length(Nb_thr_no_input, _, Nb_thr, Nb_no_input),
    constraints_CP_SAT :create_State_CP_SAT(State_f, Nb_thr_no_input, Cnf1),
    focal_state :focal_state(Eqs, Species, State_i, P_Input, State_f, Cnf2),
    transition_rule(tr(State_i, State_s), State_f, Cnf3),
    append([Cnf1, Cnf2, Cnf3], Cnf).

/*
*/
transition_rule(T, State_f, Cnf) :-
    T = tr(State_i, State_s),
    constraints_CP_SAT :create_bool_CP_SAT(Bst),
    adjacent_lists(State_i, State_s, Cnf1),
    positive_and_negative_change(State_i, State_s, LB_down, LB_up, Cnf2),
    no_change(State_i, State_s, Bst, Cnf3),
    asynchronicity(LB_down, LB_up, Bst, Cnf4),
    positive_and_negative_trend(State_i, State_f, LB_down, LB_up, Cnf5),
    no_change(State_i, State_f, Bst, Cnf6),
    lists :append([Cnf1, Cnf2, Cnf3, Cnf4, Cnf5, Cnf6], Cnf).

/*
*/
adjacent_lists([], [], []).
adjacent_lists([[IE1,E1]|L1], [[IE2,E2]|L2], Cnf) :-
    E1 - 1 #=< E2,
    cnf :trans_InfEqualXm1Y(IE1, IE2, Cnf1),
    E2 #=< E1+1,
    cnf :trans_InfEqualXm1Y(IE2, IE1, Cnf2),
    adjacent_lists(L1, L2, Cnf3),
    append([Cnf1,Cnf2,Cnf3], Cnf).

/*
*/
positive_and_negative_change([], [], [], [], []).
positive_and_negative_change([[IEi,Ei]|State_i], [[IEs,Es]|State_s], [[IBd,Bd]|LB_down], [[IBu,Bu]|LB_up], Cnf) :-
    constraints_CP_SAT :create_bool_CP_SAT([IBd,Bd]),
    constraints_CP_SAT :create_bool_CP_SAT([IBu,Bu]),
    Bd #<=> (Es #= Ei - 1),
    cnf :trans_EqualXm1YB(IEi, IEs, IBd, Cnf1),
    Bu #<=> (Es #= Ei + 1),
    cnf :trans_EqualXp1YB(IEi, IEs, IBu, Cnf2),
    positive_and_negative_change(State_i, State_s, LB_down, LB_up, Cnf3),
    append([Cnf1, Cnf2, Cnf3], Cnf).

/*
*/
no_change([], [], [IBst,Bst], Cnf) :-
    cnf :trans_EquivXcst(IBst, 1, Cnf),
    Bst #= 1.

no_change([[IE1,E1]|State1], [[IE2,E2]|State2], [IBst,Bst], Cnf) :-
    constraints_CP_SAT :create_bool_CP_SAT([IB1,B1]),
    constraints_CP_SAT :create_bool_CP_SAT([IB2,B2]),
    Bst #<=> (B1 #/\ B2),
    cnf :trans_AndXYB(IB1, IB2, IBst, Cnf1),
    B1 #<=> (E1 #= E2),

```

```

cnf :trans_EqualXYB(IE1, IE2, IB1, Cnf2),
no_change(State1, State2, [IB2,B2], Cnf3),
append([Cnf1, Cnf2, Cnf3], Cnf).
100

/*
*/
positive_and_negative_trend([], [], [], [], []).
positive_and_negative_trend([[IEi,Ei]|State_i], [[IEf,Ef]|State_f], [[IBd,Bd]|LB_down], [[IBu,Bu]|LB_up], Cnf) :-
    Bd #=> (Ei #> Ef),
    cnf :trans_SupXYIfB(IEi, IEf, IBd, Cnf1),
    Bu #=> (Ei #< Ef),
    cnf :trans_SupXYIfB(IEf, IEi, IBu, Cnf2),
    positive_and_negative_trend(State_i, State_f, LB_down, LB_up, Cnf3),
    append([Cnf1, Cnf2, Cnf3], Cnf).
110

/*
*/
asynchronicity(LB_down, LB_up, [IBst,VBst], Cnf) :-
    catch_list_var_bool(LB_down, LVB_down),
    catch_list_var_bool(LB_up, LVB_up),
    append([[VBst],LVB_down,LVB_up], LVB),
    clpfd :sum(LVB, #=, 1), % exactly one element of LVB is true
    catch_list_idf_bool(LB_down, LIB_down),
    catch_list_idf_bool(LB_up, LIB_up),
    append([[IBst],LIB_down,LIB_up], LIB),
    cnf :convert_list_idf_to_cnf(LIB, LIBcnf),
    cnf :exactly_one(LIBcnf, Cnf).
120

/*
*/
catch_list_idf_bool([], []).
catch_list_idf_bool([[IB,_] | LB], [IB|LIB]) :-
    catch_list_idf_bool(LB, LIB).
130

catch_list_var_bool([], []).
catch_list_var_bool([[_,B] | LB], [B|LVB]) :-
    catch_list_var_bool(LB, LVB).

```

F.2 Fichier Prolog du module *focal_state*

```

:- module(focal_state, [focal_state/6]).

:- use_module(library(lists)).
:- use_module(library(clpfd)).
:- use_module(cnf).
:- use_module(constraints_CP_SAT).

/*
focal_state(Eqs, Species, State_i, P_Input, State_f, Cnf) is true if
* State_f is the focal state of State_i according to
. the focal equation Eqs,
. the input concentrations P_Input, and

```

```

    . the list of species' identifiers Species,
    * Cnf is the list of clauses equivalent to the satisfiability of
    the previous statement
*/
focal_state(Eqs, Species, State_i, P_Input, State_f, Cnf) :-
    append([State_i, P_Input], State_i_o_P_Input),
    focal_state(Eqs, Species, State_i_o_P_Input, State_f, Cnf).
20

focal_state([], -, -, [], []).
focal_state([Eq|Eqs], Species, State_i_o_P_Input, [[ISf_C,Sf_C]|State_f], Cnf) :-
    Eq = eq(_C, L_P_K_C, LI_C, LV_C),
    focal_C(L_P_K_C, LV_C, LI_C, Species, State_i_o_P_Input, [ISf_C,Sf_C], Cnf1),
    focal_state(Eqs, Species, State_i_o_P_Input, State_f, Cnf2),
    append(Cnf1, Cnf2, Cnf).

/*
*/
focal_C([], [], -, -, -, []).
focal_C([[IK,K]|L_P_K_C], [V|LV_C], LI_C, Species, State_i_o_P_Input, [ISf_C,Sf_C], Cnf) :-
    constraints_CP_SAT :create_bool_CP_SAT([ICond,Cond]),
    create_def_Cond(V, LI_C, Species, State_i_o_P_Input, [ICond,Cond], Cnf1),
    Cond #=> (Sf_C #= K),
    cnf :trans_EqualXYfB(ISf_C, IK, ICond, Cnf2),
    focal_C(L_P_K_C, LV_C, LI_C, Species, State_i_o_P_Input, [ISf_C,Sf_C], Cnf3),
    append([Cnf1,Cnf2,Cnf3], Cnf).
30

/*
*/
create_def_Cond([], [], -, -, [IB,B], Cnf) :-
    B #= 1,
    cnf :trans_EquivXcst(IB, 1, Cnf).
create_def_Cond([V_J|V], [L_C_J|LI_C], Species, State_i_o_P_Input, [IB,B], Cnf) :-
    constraints_CP_SAT :create_bool_CP_SAT([IB1,B1]),
    constraints_CP_SAT :create_bool_CP_SAT([IB2,B2]),
    B #<=> (B1 #/\ B2),
    cnf :trans_AndXYB(IB1, IB2, IB, Cnf1),
    create_def_Cond_V_J(V_J, L_C_J, Species, State_i_o_P_Input, [IB1,B1], Cnf2),
    create_def_Cond(V, LI_C, Species, State_i_o_P_Input, [IB2,B2], Cnf3),
    append([Cnf1,Cnf2,Cnf3], Cnf).
50

create_def_Cond_V_J(V_J, L_C_J, Species, State_i_o_P_Input, [IB,B], Cnf) :-
    V_J = 0,
    !,
    L_C_J = i(C2, [IThr,Thr]),
    lists :nth1(N, Species, C2),
    lists :nth1(N, State_i_o_P_Input, [IConcentration,Concentration]),
    B #<=> (Concentration #< Thr),
    cnf :trans_InfXYB(IConcentration, IThr, IB, Cnf).
create_def_Cond_V_J(V_J, L_C_J, Species, State_i_o_P_Input, [IB,B], Cnf) :-
    V_J = 1,
    L_C_J = i(C2, [IThr,Thr]),
    lists :nth1(N, Species, C2),
    lists :nth1(N, State_i_o_P_Input, [IConcentration,Concentration]),
    B #<=> (Concentration #>= Thr),
    cnf :trans_SupEqualXYB(IConcentration, IThr, IB, Cnf).
70

```


Annexe G

Mise en œuvre du lien entre variables PLC et variables SAT

Nous présentons dans cette Annexe les aspects techniques de la coopération d'un solveur SAT et du solveur de contraintes PLC dans l'outil GNBox présentée en Section 8.3.

Pour réaliser le lien entre les variables PLC et leurs ensembles de variables SAT associées nous utilisons deux *variables globales* Prolog et un *prédicat dynamique*. Une variable globale correspond à une variable au sens de la programmation impérative, où les variables sont instanciées après l'initialisation et peuvent changer de valeur par la suite de l'exécution séquentielle du programme. Un prédicat dynamique est un prédicat dont la définition peut être modifiée au cours d'une exécution.

La variable globale *isat* conserve l'identifiant entier maximal courant des variables SAT. Elle est initialisée à 0 et est incrémentée de 1 pour chaque identifiant de variable SAT créé. La variable globale *ifd* conserve l'identifiant entier maximal courant des variables entière PLC qui n'ont pas encore d'identifiant à leur création. Elle est initialisée à 0 et est incrémentée de 1 pour chaque identifiant de variables PLC créé. Le prédicat dynamique $var_lk(Idf_PLC, LIdf_SAT)$ est vrai si *Idf_PLC* est un entier identifiant une variable PLC et *LIdf_SAT* est une liste d'entiers dont l'élément d'indice *i* est l'identifiant de la variable SAT associée à la *i^{eme}* valeur de la variable PLC. Une règle du prédicat $var_lk(Idf_PLC, LIdf_SAT)$ est ajoutée à chaque nouveau lien entre un identifiant *Idf_PLC* de variable PLC et sa liste associée *LIdf_SAT* d'identifiants de variables SAT.

Exemple G.10 *Considérons un ensemble de contraintes réduit à l'unique relation $B \Leftrightarrow X = Y$ avec les domaines $\{0, 1\}$, $\{0, 1, 2\}$ $\{0, 1, 2\}$ pour B , X et Y respectivement. De plus considérons que seul X possède un identifiant, qui est égal à l'atome x (il est donc nécessaire de construire deux identifiants de variables PLC, une pour B et une pour Y). Au départ $isat = 0$ et $ifd = 0$.*

L'algorithme construit un identifiant pour B en incrémentant ifd de 1, crée un identifiant SAT de Booléen pour B en incrémentant $isat$ de 1 et associe l'identifiant de B à l'identifiant de sa variable SAT en ajoutant une règle $var_lk(1, [1])$ au prédicat var_lk . A ce moment de l'exécution de l'algorithme $ifd = 1$ et $isat = 1$.

Pour la variable X de domaine $\{0, 1, 2\}$ l'algorithme construit trois identifiants de variables SAT en incrémentant $isat$ et ajoute la règle $var_lk(x, [2, 3, 4])$. A ce moment de l'exécution de l'algorithme $ifd = 1$ et $isat = 4$.

Pour la variable Y de domaine $\{0, 1, 2\}$ l'algorithme construit un identifiant pour Y en incrémentant ifd , crée trois identifiants de variables SAT en incrémentant $isat$ et ajoute la règle

$var_lk(2, [5, 6, 7])$. A ce moment de l'exécution de l'algorithme $idf = 2$ et $isat = 7$.

Ensuite l'algorithme construit les clauses suivantes placées dans une variable Prolog Cnf représentée comme une liste de listes d'identifiants de variables SAT ou de négation, noté $-$, d'identifiants de variables SAT où chacune des listes de la liste Cnf représente une clause (disjonction de littéraux) :

$$Cnf = [\quad [-1, -2, 5], [-1, 2, -5], \quad [-1, -3, 6], [-1, 3, -6], \quad [-1, -4, 7], [-1, 4, -7], \\ \quad [1, -2, -5], \quad \quad [1, -3, -6], \quad \quad [1, -4, -7] \quad]$$

Le modèle cnf fournit les prédicats nécessaires pour initialiser et mettre à jour les variables globales $isat$ et idf et le prédicat dynamique var_lk , mais aussi les prédicats pour créer des termes Prolog représentant les clauses créées pour tous les types de relations nécessaires.

Index

Numéros de pages des premières apparitions des notions essentielles :

activité d'une composition d'interaction, 26

activité d'une interaction, 16

chemin, 33

comportement, 4

composante focale, 18

composition d'interactions, 26

composition de contextes cellulaires, 27

contexte cellulaire, 16

diamètre d'un réseau, 49

ensemble \mathcal{P} de paramètres, 4

ensemble \mathcal{T} de transitions, 4

entrée, 44

état, 4

état focal, 19

état singulier, 9

graphe \mathcal{G} d'interactions, 4

interaction, 4

modèle \mathcal{M} , 4

paramètre cinétique discret, 18

paramètre d'entrée, 44

paramètre de seuil, 45

Bibliographie

- [1] M. Adélaïde and G. Sutre. Parametric analyses and abstraction of genetic regulatory networks. In Elsevier, editor, *Proceedings 2nd Workshop on Concurrent Models in Molecular Biology (BioCONCUR'04)*, 2004.
- [2] J. Ahmad, G. Bernot, J.-P. Comet, D. Lime, and O. Roux. Hybrid modelling and dynamical analysis of gene regulatory networks with delays. *ComplexUs*, 3(4), 2006.
- [3] J. Ahmad, O. Roux, G. Bernot, J.-P. Comet, and A. Richard. Analysing formal models of genetic regulatory networks with delays : Applications to lambda phage and T-cell activation systems. *Int. J. Bioinformatics Research and Applications*, 4(3) :240–262, 2008.
- [4] K. Apt. *Principles of Constraint Programming*. Cambridge University Press, 2003.
- [5] G. Batt, R. Casey, H. de Jong, J. Geiselmann, 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. *Mathematical and Computational Methods in Biology*, pages 206–239, 2006.
- [6] G. Batt, H. de Jong, and J. Geiselmann. Symbolic reachability analysis of genetic regulatory networks using discrete abstractions. *Automatica*, 44(4) :982–989, 2008.
- [7] G. Batt, D. Ropers, H. de Jong, J. Geiselmann, 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, *Proceedings Eighth International Workshop on Hybrid Systems : Computation and Control (HSCC 2005)*, LNCS 3414, pages 134–150. Springer, 2005.
- [8] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. A fruitful 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.
- [9] L. Bordeaux, Y. Hamadi, and L. Zhang. Propositional satisfiability and constraint programming : A comparative survey. *ACM Comput. Surv.*, 38(4), 2006.
- [10] L. Calzone, F. Fages, and S. Soliman. Biocham : an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14) :1805–1807, 2006.
- [11] A. Di Cara, A. Garg, G. De Micheli, I. Xenarios, and L. Mendoza. Dynamic simulation of regulatory networks using squad. *BMC Bioinformatics*, 8 :462, 2007.
- [12] N. Chabrier and F. Fages. Symbolic model checking of biochemical networks. In C. Priami, editor, *Computational Methods in System Biology*, LNCS 2602, pages 149–162. Springer, 2003.
- [13] C. Chaouiya, E. Remy, and D. Thiéffry. Qualitative petri net modelling of genetic networks. *Lecture Notes in Computer Sciences*, 4220 :95–112, 2006.

- [14] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. Nusmv version 2 : An opensource tool for symbolic model checking. In *Proc. of the 14th International Conference on Computer-Aided Verification (CAV 2002)*, volume 2404 of *LNCS*, pages 359–364. Springer-Verlag, 2002.
- [15] O. Cinquin and J. Demongeot. Positive and negative feedback : striking a balance between necessary antagonists. *J Theor Biol*, 216(2) :229–241, 2002.
- [16] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronisation skeletons using branching-time temporal logic. In *Logic of Programs, Workshop*, volume 131, pages 52–71. Springer-Verlag, 1981.
- [17] J. Cohen. Classification of approaches used to study cell regulation : Search for a unified view using constraints and machine learning. *Electronic Transactions in Artificial Intelligence, Machine Intelligence 18. Linköping Electronic Articles in Computer and Information Science ISSN*, 6(25), 2001.
- [18] F. Corblin. Inférence et simulation de réseaux biologiques logiques à l’aide de la programmation logique avec contraintes (PLC). Rapport de magistère d’informatique 3^{ème} année, Université Joseph-Fourier, Grenoble, 2005.
- [19] F. Corblin, L. Bordeaux, Y. Hamadi, E. Fanchon, and L. Trilling. A sat-based approach to decipher gene regulatory networks. In *Integrative Post-Genomics, RIAMS’07*, Lyon, 2007.
- [20] F. Corblin, E. Fanchon, and L. Trilling. Constraint logic programming for modeling a biological system described by a logical network. In *CP05 (International Conference on Principles and Practice of Constraint Programming), Workshop WCB05 (Workshop on Constraint Based Methods for Bioinformatics)*, Sitges Espagne, 2005.
- [21] F. Corblin, E. Fanchon, and L. Trilling. Inférer et simuler un modèle biologique intervenant dans l’adhérence entre cellules. In *JFPC 05 (Journées Francophones de Programmation par Contraintes)*, Lens, 2005.
- [22] F. Corblin, E. Fanchon, and L. Trilling. Modélisation d’un réseau contrôlant l’adhésion cellulaire sous la forme d’un réseau multivalué asynchrone : mise en œuvre en plc. In *JOBIM 2005 (Journées Ouvertes Biologie Informatique Mathématiques)*, Lyon, 2005.
- [23] F. Corblin, E. Fanchon, and L. Trilling. Modélisation de réseaux biologiques discrets en programmation logique par contraintes. *TSI (Technique et Science Informatiques)*, 26 :73–98, 2007.
- [24] 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.
- [25] H. de Jong and D. Ropers. Qualitative approaches towards the analysis of genetic regulatory networks. In J. Stelling (eds) Z. Szallasi, V. Periwal, editor, *System Modelling in Cellular Biology : From Concepts to Nuts and Bolts*, pages 125–148. MIT Press, Cambridge, MA, 2006.
- [26] H. de Jong, D. Ropers, C. Chaouiya, and D. Thieffry. Modélisation, analyse et simulation de réseaux de régulation génique. *Biofutur*, 252 :36–40, 2005.
- [27] V. Devloo, P. Hansen, and M. Labbé. Identification of all steady states in large biological systems by logical analysis. *Bulletin of Mathematical Biology*, 65(6) :1025–1051, 2003.
- [28] N. Eén and N. Sörensson. An extensible SAT-solver. In *SAT*, pages 502–518, 2003.

-
- [29] 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. Schachter, editors, *Computational Methods in Systems Biology*, volume 3082, pages 104–118, 2005.
- [30] E. Giunchiglia, J. Lee, V. Lifschitz, N. McCain, and H. Turner. Nonmonotonic causal theories. *Artificial Intelligence*, 153(1-2) :49–104, 2004.
- [31] A. González, C. Chaouiya, and D. Thieffry. Dynamical analysis of the regulatory network defining the dorso-ventral boundary of the drosophila wing imaginal disc. *Genetics*, 174(3) :1625–34, 2006.
- [32] G. Gössler. Compositional reachability analysis of genetic networks. In *Computational Methods in Systems Biology*, volume 4210 of *LNCS*, pages 212–226. Springer Berlin/Heidelberg, 2006.
- [33] J. Jaffar and M.J. Maher. Constraint logic programming : A survey. *Journal of Logic Programming*, (19-20) :503–581, 1994.
- [34] H. Öktem, R. Pearson, and K. Egiazarian. An adjustable aperiodic model class of genomic interactions using continuous time boolean networks (boolean delay equations). *Chaos*, 13(4) :1167–1174, 2003.
- [35] D. Mateus, J.-P. Gallois, J.-P. Comet, , and P. Le Gall. Symbolic modeling of genetic regulatory networks. *Journal of Bioinformatics and Computational Biology*, 5(2b) :627–640, 2007.
- [36] L. Mendoza and I. Xenarios. A method for the generation of standardized qualitative dynamical systems of regulatory networks. *Theoretical Biology and Medical Modelling*, 3 :13, 2006.
- [37] A. Naldi, D. Thieffry, and C. Chaouiya. Decision diagrams for the representation and analysis of logical models of genetic networks. 4695 :233–247, 2007.
- [38] M. R. Prasad, A. Biere, and A. Gupta. A survey of recent advances in SAT-based formal verification. *STTT*, 7(2) :156–173, 2005.
- [39] J.-P. Queille and J. Sifakis. Specification and verification of concurrent systems in cesar. In Springer-Verlag, editor, *Proceedings of the 5th Colloquium on International Symposium on Programming*, volume 137, pages 337–351, 1982.
- [40] 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.
- [41] A. Richard, J.-P. Comet, and G. Bernot. Formal methods for modeling biological regulatory networks. In Gabbar Hossam A., editor, *Modern Formal Methods and Applications*. Springer, 2006.
- [42] D. Ropers, H. de Jong, M. Page, D. Schneider, and J. Geiselman. Qualitative simulation of the carbon starvation response in *Escherichia coli*. *Biosystems*, 84(2) :124–152, 2006.
- [43] L. Sánchez and D. Thieffry. A logical analysis of the *Drosophila* gap-gene system. *J. theor. Biol.*, 211 :115–141, 2001.
- [44] H. Siebert and A. Bockmayr. Temporal constraints in the logical analysis of regulatory networks. *Theoretical Computer Science*, 391(3) :258–275, 2008.
- [45] E. H. Snoussi. Qualitative dynamics of piecewise-linear differential equations : A discrete mapping approach. *Dynamics and Stability of Systems*, 4 :189–207, 1989.

- [46] E. H. Snoussi and R. Thomas. Logical identification of all steady states : The concept of feedback loop characteristic states. *Bulletin of Mathematical Biology*, 55 :973–991, 1993.
- [47] L. J. Steggles, R. Banks, O. Shaw, and A. Wipat. Qualitative modelling and analysing genetic regulatory networks : a petri net approach. *Bioinformatics*, 23(3) :336–343, 2007.
- [48] M. Thellier, J. Demongeot, V. Norris, J. Guespin, C. Ripoll, and R. Thomas. A logical (discrete) formulation for the storage and recall of environmental signals in plants. *Plant biol*, 6(5) :590–597, 2004.
- [49] D. Thieffry, M. Colet, and R. Thomas. Formalisation of regulatory networks : a logical method and its automation. *Math. Modelling and Sci. Computing*, 2 :144–151, 1993.
- [50] D. Thieffry and R. Thomas. Dynamical behaviour of biological regulatory networks – ii. immunity control in bacteriophage lambda. *Bulletin of Mathematical Biology*, 57 :277–297, 1995.
- [51] R. Thomas and R. D’Ari. *Biological Feedback*. CRC Press, 1990.
- [52] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. ii. logical analysis of regulatory networks in term of feedback circuits. *Chaos*, 11 :180–195, 2001.
- [53] 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 :247–276, 1995.
- [54] S. Troncale, J.-P. Comet, and G. Bernot. Verification of biological models with timed hybrid petri nets. In T. D. Pham and X. Zhou, editors, *Proceedings of the 2007 Intl Symp on Computational Models for Life Sciences (CMLS07)*, volume 952, pages 287–296, 2007.
- [55] D. Weinstein-Fischer and S. Altuvia. Differential regulation of escherichia coli topoisomerase i by fis. *Mol. Microbiol.*, 63(4) :1131–1144, 2007.