



HAL
open science

Modèles hybrides de réseaux de régulation : étude du couplage des cycles cellulaire et circadien

Jonathan Behaegel

► **To cite this version:**

Jonathan Behaegel. Modèles hybrides de réseaux de régulation : étude du couplage des cycles cellulaire et circadien. Bio-informatique [q-bio.QM]. COMUE Université Côte d'Azur (2015 - 2019), 2018. Français. NNT : 2018AZUR4071 . tel-01962051

HAL Id: tel-01962051

<https://theses.hal.science/tel-01962051v1>

Submitted on 20 Dec 2018

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.



THÈSE DE DOCTORAT

Modèles hybrides de réseaux de régulation : étude du couplage des cycles cellulaire et circadien

Présentée en vue de l'obtention du grade de docteur en Informatique de
l'Université Côte d'Azur par

Jonathan BEHAEGEL

Laboratoire I3S, UMR7271, CNRS
Laboratoire IBV, UMR7277, CNRS, Inserm U1091

Dirigée par : Jean-Paul Comet, Franck Delaunay
Co-encadrée par : Marie Pelleau

Soutenue le 02 Octobre 2018

devant le jury composé de :

M. Alexander	BOCKMAYR	Professeur	Université de Berlin	Rapporteur
M. Jean-Paul	COMET	Professeur	Université de Nice	Directeur de thèse
M. Hidde	DE JONG	Directeur de Recherches	INRIA Grenoble	Rapporteur
M. Franck	DELAUNAY	Professeur	Université de Nice	Directeur de thèse
Mme. Cathy	ESCAZUT	Maître de Conférences	Université de Nice	Membre invitée
M. Michel	GAUTERO	Maître de Conférences	Université de Nice	Membre invité
M. Francis	LÉVI	Professeur	Université de Warwick	Examinateur
Mme. Marie	PELLEAU	Maître de Conférences	Université de Nice	Co-encadrante
Mme. Sabine	PERES	Maître de Conférences	Université Paris-Sud	Examinatrice

Modèles hybrides de réseaux de régulation : étude du couplage des cycles cellulaire et circadien

Jury :

Rapporteurs

M. Alexander BOCKMAYR,
M. Hidde DE JONG,

Professeur
Directeur de Recherches

Université de Berlin
INRIA Grenoble

Examineurs

M. Francis LÉVI,
Mme. Sabine PERES,

Professeur,
Maître de Conférences

Université de Warwick
Université Paris-Sud

Invités

Mme. Cathy ESCAZUT,
M. Michel GAUTERO,

Maître de Conférences
Maître de Conférences

Université de Nice
Université de Nice

Directeurs de thèse et co-encadrante

M. Jean-Paul COMET,
M. Franck DELAUNAY,
Mme. Marie PELLEAU,

Professeur
Professeur
Maître de Conférences

Université de Nice
Université de Nice
Université de Nice

Modèles hybrides de réseaux de régulation : étude du couplage des cycles cellulaire et circadien

Résumé

La modélisation de systèmes biologiques est devenue indispensable pour comprendre les phénomènes complexes et émergents issus d'influences partiellement connues, et pour envisager de contrôler un système altéré dans le but de restaurer un comportement physiologique. Tout modèle, quelque soit son paradigme sous-jacent, fait intervenir des paramètres gouvernant sa dynamique mais les mesures expérimentales ne permettent généralement pas de les identifier et cela reste l'un des problèmes majeurs de la modélisation. Cette thèse propose une méthode automatique d'identification des paramètres dynamiques de systèmes biologiques dans un cadre de modélisation hybride.

Le cadre hybride choisi découpe l'espace des phases selon l'activité des entités biologiques, et associe à chacun de ces sous-espaces une vitesse d'évolution de chacun des composants. Nous proposons une logique de Hoare en temps continu ainsi qu'un calcul de plus faible précondition qui, à partir d'observations expérimentales qualitatives et chronométriques, construit les contraintes minimales sur les paramètres du modèle pour qu'il soit compatible avec les observations. Ce calcul mène à un problème de satisfaction de contraintes sur les réels et nous montrons que celui-ci peut être résolu par le solveur AbSolute.

Le prototype Holmes BioNet développé au cours de cette thèse peut non seulement automatiser le processus d'identification des valeurs des paramètres à partir des observations expérimentales, mais aussi simuler l'évolution du modèle obtenu afin de le comparer avec les traces expérimentales. Nous utilisons ce prototype pour modéliser le couplage des cycles cellulaire et circadien.

Mots clés : Réseaux de régulation biologique, Logique de Hoare hybride, Contraintes continues, Cycles cellulaire et circadien

Hybrid models of regulatory networks : a study of cellular and circadian cycles coupling

Abstract

Modelling biological systems has become instrumental to understand complex and emerging phenomena resulting from partially known influences, and to consider controlling an altered system in order to restore a physiological behaviour. Any model, independent of the underlying paradigm, involves parameters governing its dynamics. However, experimental measurements generally do not allow their identification and this remains one of the major problems of modelling. This PhD proposes an automatic method for identifying the dynamic parameters of biological systems in a hybrid modelling framework.

The chosen hybrid framework splits the phase space according to the activity of the biological entities, and associates to each of these subspaces a celerity for each of the components. We introduce a continuous time Hoare logic as well as its weakest precondition calculus which, from qualitative and chronometrical experimental observations, constructs the minimum constraints on the model parameters making it compatible with the observations. This calculus leads to a Constraint Satisfaction Problem on real numbers and we show that it can be solved by the AbSolute solver.

The Holmes BioNet prototype developed during this PhD can not only automate the parameter identification process from experimental data, but also simulate the evolution of the obtained model in order to compare it with experimental traces. We use this prototype to model the coupling of the cellular and circadian cycles.

Keywords : Biological regulatory networks, Hybrid Hoare logic, Continuous constraints, Cellular and circadian cycles

Remerciements

Tout d'abord, je tenais à remercier les membres du jury de leur présence et leur participation à ma thèse, et plus particulièrement mes deux rapporteurs, Alexander Bockmayr et Hidde de Jong.

Je tenais ensuite à remercier mes directeurs de thèse, Jean-Paul Comet et Franck Delaunay pour leur bienveillance et leurs aides précieuses durant ces trois années. Merci également à Marie Pelleau pour son encadrement et sans qui les solveurs de contraintes seraient toujours très abstraits!

Je remercie également les différentes équipes des deux laboratoires, toute l'équipe MDSC (notamment Cinzia Di Giusto, Elisabetta De Maria, Joëlle Despeyroux, Adrien Richard), tous les membres du projet Bioinfo Formelle et MS&N (Alexandre Muzy, Gilles Bernot, Patrick Coquillard), toute l'équipe Circadian System Biology (Aline Grechez-Cassiau, Michèle Teboul, Sophie Guérin), toutes les personnes rencontrées à l'iBV et merci à Agnès Loubat pour son aide dans les expériences réalisées. Merci à Magali Richir pour toute son aide au secrétariat malgré les péripéties de début et de fin de thèse.

Je remercie également tous les collègues rencontrés durant ces trois ans à l'I3S et l'iBV et aussi durant les études : merci à Déborah, Eman, Ingrid et Ophélie (team GB), Perrine, Anthony, Cyril, Heytem (team 114), John, Oussama, Pablo, Pierre, Rajeev, Rémy (merci pour tous les bons gâteaux!). Merci pour toutes les discussions pendant le repas et autour du café. Un grand merci également à Benjamin (même quand tu me disais qu'on avait fini de courir), Emilien, Guillaume et Maxime.

Merci à Fernando et Ninad (los 2 compadres) pour avoir vécu plus d'un an génial à vos côtés ainsi qu'avec les autres compadres (Caroline, Myriana, Xiomhara, Majdi). A mes amis de longue date, merci à Camille, Claudia, Cyril, Morgan, Steve et tous les autres amis que je n'ai pas cités.

A ma famille pour leur soutien permanent, et plus particulièrement à mes parents pour avoir cru en moi durant toutes ces années d'études.

Et bien entendu, à Mireille, pour son soutien infailible dans les moments les plus difficiles de la thèse. Place à nos projets!

Table des matières

1	Introduction	11
2	Deux cycles régulateurs chez les mammifères et leurs modélisations	19
2.1	Description biologique	19
2.1.1	Cycle cellulaire	20
	Les phases du cycle cellulaire	21
	Mécanismes de surveillance	24
	Approche mécanistique	24
2.1.2	Cycle circadien	30
	Caractéristiques générales	30
	Approche mécanistique	31
2.2	Les cadres de modélisation	32
2.3	Modèles existants	35
2.3.1	Les modèles du cycle cellulaire	35
2.3.2	Les modèles du cycle circadien	38
2.3.3	Les modèles couplant les deux cycles biologiques	39
2.4	Vers un nouveau modèle du couplage	40
3	Définition du cadre de modélisation utilisé	43
3.1	Motivation	43
3.2	Le cadre de modélisation de René Thomas avec multiplexes	44
3.3	Le cadre de modélisation hybride	48
3.3.1	Réseau de régulation hybride	48
3.3.2	Etat d'un réseau de régulation hybride	49
3.3.3	Délai de contact	50
3.3.4	Murs interne et externe	52
3.3.5	Entité changeante <i>first</i>	53

TABLE DES MATIÈRES

3.3.6	Graphe d'états hybrides : transitions continues et discrètes	53
3.4	Exemple sur la boucle négative	55
3.5	Positionnement par rapport à nos objectifs	57
4	Construction des contraintes sur les paramètres	59
4.1	Vérification des automates hybrides	59
4.1.1	Rappel sur les automates hybrides et les méthodes formelles	59
4.1.2	Classes d'automates hybrides et décidabilité/indécidabilité	62
	Les automates hybrides linéaires	62
	Les automates hybrides rectangulaires	62
	Système dérivé constant par morceau	65
4.1.3	Les outils de vérification existants	65
4.2	Logique de Hoare pour la vérification de modèles discrets	66
4.2.1	Vérification d'un programme	66
4.2.2	Logique de Hoare génétiquement modifiée	68
4.3	Logique de Hoare hybride	69
4.3.1	Langage de propriétés	69
4.3.2	Langage de chemins	70
4.3.3	Triplets de Hoare et leur sémantique	74
4.3.4	Calcul de la plus faible précondition	76
4.3.5	Règles d'inférence	82
4.4	Preuve de correction	83
4.4.1	Deux lemmes	84
4.4.2	Preuve de la correction par induction	85
	Premier cas : $a = \top$	85
	Deuxième cas : $a = \text{slide}^+(u)$ ou $a = \text{slide}^-(u)$ avec $u \neq v$	86
	Troisième cas : $a = \text{noslide}^+(u)$ ou $a = \text{noslide}^-(u)$	87
	Quatrième cas : $a = C_u \square c$ où $c \in \mathbb{R}$ et \square est l'un des comparateurs habituels	88
	Cinquième cas : $a = a_1 \wedge a_2$	88
4.5	Complétude sémantique et preuve	88
4.6	Le cycle négatif développé pas à pas	92
5	Résolution des contraintes grâce à un solveur	97
5.1	Introduction sur les méthodes de solveurs	97
5.2	Premier essai avec Ibex	102

5.3	Les avancées grâce au solveur AbSolute	103
5.3.1	Réécriture symbolique	104
5.3.2	Domaines de définition invariants et changeants	104
5.3.3	Limites du solveur	106
5.4	Utilisation d'AbSolute sur le cycle négatif	106
5.4.1	Comportement cyclique du cycle négatif	108
5.4.2	Comportement acyclique du cycle négatif	108
5.4.3	Comportement cyclique du cycle négatif avec temps Δt inconnus	109
5.4.4	Comportements cycliques de rétrocontrôle négatif de tailles différentes	109
6	Le prototype Holmes BioNet	113
6.1	Données d'entrée : Graphe d'influence et trace biologique	113
6.1.1	Ecriture du graphe d'influence dans le fichier d'entrée	115
6.1.2	Ecriture du triplet de Hoare dans le fichier d'entrée	116
6.2	Construction des contraintes	118
6.3	Appel au solveur AbSolute	120
6.4	Simulateur	121
6.5	Lancement du prototype et résultats	123
6.5.1	Cycle circadien	123
6.5.2	Cycle cellulaire	125
7	Vers une compréhension du couplage des cycles cellulaire et circadien	131
7.1	Graphe d'influence	131
7.2	Verrouillage des phases des deux cycles	132
7.3	Expériences	135
7.4	Construction du triplet de Hoare	139
7.5	Utilisation du prototype et simulation	140
7.6	Discussion	140
8	Conclusion	143
8.0.1	Contributions	143
8.0.2	Perspectives	145
9	Annexes	149
9.1	Contraintes du rétrocontrôle négatif	149
9.2	Etude préliminaire de l'influence du nombre d'entité sur le temps de résolution.	150
9.3	Cycle circadien simplifié en fonction du cycle jour-nuit	151

TABLE DES MATIÈRES

9.4	Simulation du cycle circadien à 3 entités	152
9.5	Evolution du graphe d'influence à 5 entités du cycle cellulaire	152

Chapitre 1

Introduction

Depuis plusieurs décennies, les résultats obtenus par des outils informatiques fiables apportent de nouvelles informations aussi prometteuses que surprenantes dans le domaine de la biologie. Ces outils, regroupés sous le terme *in silico*, sont en plein essor. Toutefois, ces méthodes ne remplacent pas les méthodes *in vitro* et *in vivo* existantes mais viennent plutôt compléter les informations obtenues. Pour citer quelques exemples de méthodes *in silico*, le *machine learning* ou apprentissage automatique développe des méthodes d'identification de composés ayant un rôle spécifique comme par exemple la découverte d'un nouveau gène de l'horloge circadienne nommé *chrono* [6], ou encore des modèles *QSAR* (Quantitative Structure-Activity Relationship) qui tentent de prédire les propriétés physico-chimiques et biologiques de molécules à partir de leurs structures tri-dimensionnelles. On peut aussi penser à la modélisation de phénomènes biologiques [99, 54, 37] menant à identifier de nouveaux mécanismes [58]. Les résultats de tels outils informatiques sont parfois difficiles à interpréter et requièrent alors une expertise pluridisciplinaire. Le bio-informaticien adapte son discours auprès des experts biologistes et informaticiens du domaine : face au biologiste, le bio-informaticien prend sa casquette d'informaticien, et son rôle de biologiste reprend le dessus devant les informaticiens.

En 2013, la commission européenne publie un rapport sur les méthodes de substitution à l'expérimentation animale dans le secteur des cosmétiques [43]. Ce rapport vise à limiter les expérimentations *in vivo* et les remplace par des méthodes alternatives provenant des techniques *in vitro* et *in silico*. L'un des défis résultant de ce rapport consiste à développer de nouveaux outils informatiques fiables. Dans l'état actuel, l'utilisation de ces outils informatiques ne peut pas se détacher totalement des expérimentations animales, mais ces outils les accompagnent pour faciliter la prise de décision et se diriger vers une expérimentation plus éthique.

Pour construire un modèle informatique d'un système biologique, il est nécessaire d'en connaître assez précisément le fonctionnement et un tel modèle sera considéré cohérent s'il reflète au moins les observations expérimentales.

Dans cette thèse, nous allons nous focaliser sur les **systèmes biologiques** moléculaires qui sont souvent définis comme un ensemble d'entités biologiques qui assure une fonction biologique. Ces systèmes biologiques sont représentés sous la forme d'un **réseau de régulation**. Ces réseaux décrivent des mécanismes biologiques entre des entités (gènes, protéines, *etc.*) dont les influences peuvent être actives ou inactives, assurant par exemple la synthèse de protéines par des gènes ou encore l'inhibition de gènes par des protéines régulatrices. Généralement, un gène s'active à un moment précis dans le processus des réactions chimiques du système biologique pour assurer sa fonction.

La complexité de systèmes biologiques provient de l'enchevêtrement des boucles de **rétrocontrôles positives et négatives**. En absence de ces rétrocontrôles, le système biologique converge vers un état stable unique, et pour sortir de cet état, il est nécessaire qu'un facteur extérieur influe sur le système. Lorsqu'ils sont présents, les rétrocontrôles négatifs mènent le système à osciller, permettant **l'homéostasie**, c'est-à-dire la capacité de maintenir les entités biologiques en quantité normale pour assurer le bon fonctionnement du système. Les rétrocontrôles positifs, quand à eux, assurent la **multistationnarité** du système lui permettant d'atteindre différents états stationnaires ou bassins d'attraction. L'attracteur vers lequel se dirige le système dépend alors de l'état de départ des différentes entités biologiques.

Ces deux types de rétrocontrôles sont largement observés dans la nature, de l'échelle moléculaire en passant par l'échelle cellulaire jusqu'à l'échelle phénotypique et même l'échelle des populations (exemple du modèle proies-prédateurs). Les rétrocontrôles positifs et négatifs permettent alors aux systèmes biologiques de s'adapter à des perturbations de l'environnement. Les éléments perturbateurs sont de natures différentes, allant des facteurs favorisant un stress cellulaire (oxydation, pH, pression osmotique, *etc.*) à des éléments étrangers à la cellule (médicament, *etc.*).

Du point de vue informatique, un réseau de régulation d'un système biologique est représenté par un **graphe d'influence** dont les sommets correspondent à une abstraction d'une ou plusieurs entités biologiques ayant le même rôle, et les arcs représentent les influences entre ces différentes entités. La **modélisation** de ces systèmes biologiques sous forme de graphe d'influence ne permet pas directement d'analyser l'évolution du système étudié car ces graphes d'influence sont statiques, mais ces modèles peuvent assister tout de même les biologistes pour interpréter les données expérimentales qu'ils peuvent obtenir, et pour comprendre à différentes échelles les comportements du phénomène étudié.

Les **cadres de modélisation** ont pour rôle de représenter l'évolution des entités présentes dans le graphe et diffèrent selon les aspects qu'ils mettent en avant.

- ◊ Les modèles différentiels décrivent, pour un réseau de régulation donné, l'évolution continue de la concentration des entités biologiques mesurée expérimentalement. Le niveau de description attendu est très élevé et les graphes d'influence sont généralement très détaillés.
- ◊ Les modèles stochastiques prennent en compte le comportement aléatoire que les mesures expérimentales peuvent révéler. Ce sont des approches probabilistes qui intègrent la variabilité des résultats expérimentaux.
- ◊ Les modèles qualitatifs s'intéressent aux caractéristiques majeures qui expliquent l'observation. Les activités des entités biologiques sont représentées par des nombres pouvant prendre un nombre fini de valeurs où chaque valeur représente un intervalle de concentration dans lequel l'effet de l'entité reste inchangé. Un des cas classiques est l'approche discrète de René Thomas où l'évolution de chaque entité biologique dépend de ses entités régulatrices [124]. Un autre cadre de modélisation qualitatif est celui des réseaux de Petri [62, 31, 102] où les places comptent les molécules de chaque espèce chimique et les transitions représentent les réactions biochimiques. Les arcs étiquetés qui relient une place à une transition spécifient les réactifs de la réaction associée à la transition alors que les arcs étiquetés reliant une transition à une place spécifient les produits de la réaction. La transition est dite franchissable si la réaction peut avoir lieu, autrement dit si toutes les places correspondant à des réactifs contiennent suffisamment de molécules. Lorsqu'une transition est franchissable, alors elle peut être franchie et dans ce cas, les réactifs vont disparaître et les produits vont apparaître.
- ◊ Les modèles hybrides combinent plusieurs cadres de modélisation. L'exemple le plus fréquent regroupe l'approche qualitative avec l'approche différentielle [57, 11].

Ces modèles, aussi divers soient-ils, sont gouvernés par des **paramètres dynamiques** qui contrôlent l'évolution des entités biologiques. Malheureusement, ces paramètres ne sont généralement pas connus et pas souvent mesurables. L'identification des paramètres est donc une problématique générale dont le but est de trouver les bonnes valeurs des paramètres pour représenter le plus fidèlement possible le comportement observé du système biologique. Dans ce contexte, comment choisir le cadre de modélisation adapté aux attentes des utilisateurs ? D'un côté, l'identification des paramètres des modèles différentiels et stochastiques relève de la recherche et l'ajout d'entités biologiques dans le graphe d'influence mène généralement à recommencer totalement tout le processus d'identification de l'ensemble des paramètres dynamiques du système. De l'autre, l'utilisation de méthodes formelles dans des modèles qualitatifs a permis d'automatiser la recherche des paramètres dynamiques [17] qui expliquent qualitativement

l'observation. Cependant, les modèles purement qualitatifs ne permettent pas de prendre en compte des informations temporelles comme par exemple les délais séparant des événements alors même que ces informations peuvent contraindre les paramètres dynamiques.

Les modèles hybrides semblent alors un bon compromis pour y inclure les avantages des modèles qualitatifs et différentiels sans prendre en compte leurs inconvénients et en intégrant les durées entre les événements.

Notre approche. Le travail présenté dans ce document contribue à l'identification des paramètres dynamiques gouvernant l'évolution de modèles hybrides. Les dynamiques de tels modèles sont représentées par des **automates hybrides**. L'identification des paramètres se base alors sur le graphe d'influence et s'appuie sur les observations expérimentales. Nous illustrons cette démarche sur le couplage du cycle cellulaire, qui est le processus de division de la cellule, avec le cycle circadien qui mène à synchroniser les cellules sur une période de 24 heures.

Les **méthodes formelles** ont été largement utilisées pour modéliser et analyser des systèmes biologiques. A l'origine, ces méthodes utilisent des logiques mathématiques dans le but de montrer qu'un programme est conforme à sa spécification. Détournées de ses fonctions premières, des techniques de **vérification formelle** ont déjà été utilisées pour vérifier des propriétés dynamiques de modèles discrets [19, 18, 45], de modèles hybrides [4] ou encore de modèles stochastiques [66]. Par exemple, dans le cadre discret, le *model checking* a été appliqué à grande échelle pour sélectionner les paramètres représentant correctement les observations biologiques décrites par des formules de logique temporelle [77]. Mais cette énumération très coûteuse peut être évitée en utilisant par exemple des outils très puissants de résolution de problèmes de décision comme ASP (pour *Answer Set Programming*) [90, 136]. Ici, nous développons un autre point de vue qui est basé sur la logique de Hoare.

La **logique de Hoare** permet de raisonner sur la correction de programmes. Introduite en 1969 par Tony Hoare [67], cette méthode repose sur des règles associées à chaque instruction syntaxique des programmes impératifs. La logique de Hoare raisonne sur la notion de **triplet de Hoare** noté $\{Pre\} p \{Post\}$ définissant des pré et postconditions d'états du système Pre et $Post$, et sur le programme p . Lorsque, à partir d'un état du système satisfaisant Pre , l'exécution du programme p est possible et termine, cette exécution mène à un état du système satisfaisant $Post$. Le triplet est alors dit correct.

Edsger Dijkstra a complété la logique de Hoare en définissant une **sémantique de transformation de prédicats** pour les langages de programmation impératifs [40] : à chaque instruction élémentaire est associée une transformation de prédicats. A chaque postcondition $Post$, et pour chaque instruction élémentaire, cette transformation de prédicats permet de construire la plus

faible précondition qui assure que l'exécution de l'instruction élémentaire termine dans un état satisfaisant *Post*. Lorsque le programme correspond à une succession d'instructions élémentaires, la plus faible précondition est calculée sur la dernière instruction et le procédé est réitéré jusqu'à identifier la plus faible précondition du programme complet.

Nous avons détourné la logique de Hoare de son rôle usuel pour qu'elle puisse être utile à l'identification des paramètres dynamiques d'un modèle hybride qui doit être *a minima* compatible avec les données issues d'observations expérimentales. Ici, les traces observées comportent non seulement la liste des événements successifs, mais aussi des informations temporelles (délais séparant deux événements consécutifs). Nous avons choisi cette méthode formelle pour contraindre des paramètres dynamiques dans des modèles hybrides car la logique de Hoare a déjà montré son intérêt pour construire de façon automatique les paramètres dynamiques de modèles discrets de systèmes biologiques [17].

Le calcul de plus faible précondition d'une succession d'événements biologiques mène à des contraintes qu'il est souvent difficile de résoudre à la main. C'est un **problème de satisfaction de contraintes** (*CSP* pour *Constraint Satisfaction Problem*). Les CSPs sont retrouvés fréquemment dans notre vie quotidienne : planification de trafic aérien, constitution d'emploi du temps dans les écoles, *etc.* La recherche de solutions de ces CSPs varie selon le problème posé. Il peut s'agir de trouver une ou plusieurs solutions satisfaisant l'ensemble des contraintes, de trouver une solution optimale minimisant un critère (température, temps, *etc.*), ou encore de prouver qu'il n'existe pas de solution. La recherche de solutions requiert l'utilisation de **solveur de contraintes**. Ces solveurs prennent en entrée le CSP décrit par l'ensemble des variables, leurs domaines de définition et les contraintes. Les solveurs tentent alors de réduire la taille de l'espace des solutions.

Ce document a pour but de présenter toutes les notions essentielles menant à identifier un ou plusieurs jeux de paramètres qui gouvernent la dynamique d'un modèle hybride représentant un système biologique à comportement oscillatoire. Il est organisé en 6 chapitres principaux.

Dans le **chapitre 2**, nous décrivons les deux phénomènes biologiques sur lesquels se basent nos travaux pour comprendre le couplage éventuel entre ces deux phénomènes à comportement oscillatoire. Dans un premier temps, nous décrivons les mécanismes intervenant dans le **cycle cellulaire** qui permettent à la cellule de se diviser. Les 4 phases principales de ce système modifient le comportement de la cellule pour démarrer la réplication de son ADN et accroître sa taille (phase *G1*), poursuivre cette réplication et dupliquer les éléments essentiels responsables

de la division de la cellule (phase S), se préparer à la division et finaliser la réplication de l'ADN (phase $G2$), et enfin assurer sa division en plusieurs étapes (phase M). Nous décrivons aussi les mécanismes de surveillance existant entre les différentes phases dont le but est de corriger les possibles erreurs de réplication.

Dans un second temps, nous décrivons le **cycle circadien** (où horloge circadienne) qui regroupe les mécanismes assurant la coordination temporelle des processus biologiques pour adapter l'organisme au **cycle nycthéméral** (alternance jour/nuit sur 24 heures). Ces mécanismes permettent de synchroniser les cellules de l'organisme *via* le comportement oscillatoire des gènes de l'horloge.

Ce chapitre présente aussi une liste non exhaustive de modèles de ces deux cycles et de leur couplage avec leurs paramètres dynamiques identifiés. On peut noter que l'évolution de ces systèmes biologiques a été modélisée dans différents cadres de modélisation (différentiels, qualitatifs, stochastiques ou hybrides) selon les informations mises en avant.

Nous proposons enfin un graphe d'influence du couplage des cycles cellulaire et circadien basé sur les connaissances biologiques actuelles.

Le **chapitre 3** présente le cadre de modélisation hybride choisi. Ce cadre de modélisation hybride se base sur l'approche discrète de René Thomas pour laquelle des espaces de concentrations des entités biologiques sont découpés en zones indexées par des valeurs qualitatives (aussi appelées niveaux d'expression) de telle manière que le comportement de chaque entité dans chaque zone est constant ou quasi constant. Ces zones aussi appelées états sont séparés par des seuils. L'approche hybride inclut également des vitesses d'évolution à l'intérieur de chaque état [37]. Ces paramètres dynamiques gouvernent alors l'évolution dans chaque état du système et sont appelées **célérités**.

Le **chapitre 4** présente les automates hybrides et positionne le modèle choisi par rapport aux différentes classes d'automates existants. Nous présentons ensuite les méthodes formelles associées et les outils existants et replaçons dans ce contexte notre problème qui nous a poussé à développer la logique de Hoare hybride. Après un rapide rappel de la logique de Hoare sur les programmes impératifs et sa modification pour les modèles discrets de réseau de régulation [17], nous détaillons son adaptation aux modèles hybrides. Les observations expérimentales sont représentées comme une succession d'événements chronométrés appelée **trace biologique**. La méthode de Dijkstra précédemment esquissée est également modifiée pour réaliser le calcul de plus faible précondition sur la trace biologique donnée. Ensuite, la correction et la complétude sémantique de la logique de Hoare hybride sont prouvées, indiquant que tout ce qui peut être prouvé avec les règles d'inférence définies est vrai du point de vue de la sémantique (correction), et que la précondition calculée pour cette trace biologique est bien la plus faible précondition (complétude sémantique).

Enfin, un exemple de construction des contraintes sur les paramètres dynamiques est développé pas à pas sur un cas classique en biologie : la boucle de rétrocontrôle négative.

Le **chapitre 5** montre comment des solveurs de contraintes sur les réels peuvent être utilisés pour résoudre le CSP issu du calcul de plus faible précondition de la logique de Hoare hybride. Les deux solveurs testés, Ibex et AbSolute, résolvent des systèmes de contraintes linéaires ou non linéaires pour des variables continues. Nous expliquons comment nous avons donné nos contraintes à de tels solveurs, quelles ont été les modifications apportées pour résoudre nos CSPs, et quelles ont été les limites. Les modifications apportées permettent notamment de réduire la taille du problème et de choisir des valeurs dans les domaines de définition des variables pour identifier, *a minima*, un jeu de paramètres.

En dernière partie de ce chapitre, nous présentons des résultats qualitatifs sur la boucle de rétrocontrôle négative et nous présentons quelques premières observations sur le temps d'exécution du solveur.

Le **chapitre 6** est consacré au prototype Holmes BioNet (*HOare Logic Modelling Experiments on Biological Networks*). A partir d'un graphe d'influence représentant un système biologique et une trace biologique écrite sous un format bien précis, Holmes BioNet calcule la plus faible précondition associée à cette trace biologique, et identifie un jeu de paramètres satisfaisant le CSP issu du calcul de plus faible précondition. Ce jeu de paramètres est ensuite utilisé pour simuler la dynamique des différentes entités du graphe d'influence. L'utilisateur pourra alors comparer cette simulation aux observations expérimentales connues.

Nous présentons l'utilisation de Holmes BioNet sur deux exemples : le cycle circadien modélisé par un graphe d'influence de 3 entités biologiques, et le cycle cellulaire comportant 5 entités.

Le **chapitre 7** s'intéresse à l'identification des paramètres dynamiques du couplage des cycles cellulaire et circadien présenté en fin de chapitre 2. Nous expliquons les éléments qui nous permettent de verrouiller les phases de ces deux cycles pour écrire une trace biologique unique combinant les traces biologiques définies dans les deux exemples du chapitre 6. Nous expliquons les expériences biologiques qui ont été réalisées pour spécifier le verrouillage des phases et proposons une trace biologique plausible. Cette trace et le graphe d'influence du couplage précédemment défini sont donnés au prototype Holmes BioNet. Nous analysons les résultats obtenus et les comparons avec les connaissances biologiques actuelles. Nous proposons aussi des améliorations en vue d'obtenir un modèle fiable.

Pour terminer, le **chapitre 8** synthétise les résultats obtenus sur l'identification des paramètres dynamiques de modèles hybrides. Nous discutons les limites du formalisme de la logique de Hoare hybride couplé au solveur AbSolute. Nous évaluons aussi les améliorations possibles, nous proposons d'étendre nos travaux de recherche menant à identifier des paramètres dynamiques à d'autres cadres de modélisation, et nous montrons comment l'application de nos

Chapitre 1 : Introduction

travaux peuvent favoriser la recherche translationnelle en cancérologie, pour la prise de médicament à l'heure la plus appropriée.

Chapitre 2

Deux cycles régulateurs chez les mammifères et leurs modélisations

En biologie, la modélisation est une aide précieuse pour l'étude des systèmes biologiques complexes et permet d'en donner des représentations à différents niveaux de description plus ou moins détaillés. L'enjeu est de confronter une ou plusieurs hypothèses aux diverses connaissances afin de les confirmer ou les infirmer. Toute connaissance des phénomènes biologiques analysés est utile et la bonne compréhension des cadres de modélisation permet de cibler le niveau de description souhaité pour pouvoir statuer sur les hypothèses d'entrées. L'objectif de ce chapitre consiste à détailler les connaissances biologiques et les cadres de modélisation utilisés par la suite. Dans un premier temps, nous décrivons le cycle cellulaire et le cycle circadien chez les mammifères. Ensuite, nous présentons les différents cadres de modélisation et nous détaillons un bon nombre de modèles déjà publiés du cycle cellulaire, du cycle circadien, et de leur couplage. Enfin, nous décrivons le modèle du couplage de ces deux cycles sur lequel se basent les résultats présentés dans les chapitres suivants.

2.1 Description biologique

Avant de s'intéresser au cycle cellulaire, il est important de comprendre l'importance du matériel génétique dans une cellule. Le **matériel génétique** détermine l'ensemble des caractères d'un organisme vivant d'un point de vue moléculaire, structural, développemental, physiologique et comportemental. Ce matériel constitué par l'**ADN** (Acide Désoxyribonucléique) est organisé en éléments fonctionnels appelés **gènes** mais contient aussi beaucoup de séquences intergéniques dont le décryptage reste inachevé. Les gènes sont transcrits en molécule d'ARN (Acide Ribonucléique) pour assurer un rôle régulateur ou pour être traduit en protéine : on

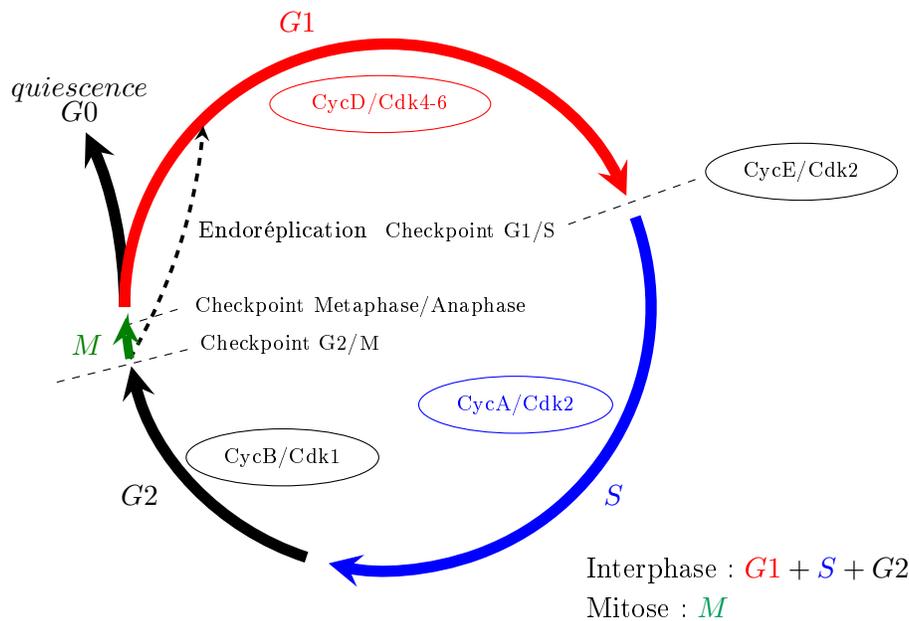


FIGURE 2.1 – Les différentes phases du cycle cellulaire. Les phases G1, S et G2 définissent l'interphase, tandis que la phase M est la phase de mitose.

parle de transcription et de traduction. Enfin, un **facteur de transcription** est une protéine initiant ou régulant la transcription d'un gène pour synthétiser une molécule d'ARN.

2.1.1 Cycle cellulaire

Le **cycle cellulaire** est un phénomène biologique qui permet à la cellule de se reproduire. Différentes étapes existent et mènent à diviser la cellule en deux cellules filles pour préserver le matériel génétique dans la descendance. Ce phénomène biologique diffère selon les organismes et la nature de leurs cellules. Par exemple, le mode de division cellulaire des cellules procaryotes dépourvues de noyau se fait par scissiparité, c'est-à-dire par simple division. À l'inverse, les cellules eucaryotes contenant un noyau présente un cycle cellulaire en deux étapes que sont **l'interphase** et **la mitose**, voir figure 2.1 Il existe aussi une phase de repos appelée **phase de quiescence** : la cellule sort temporairement ou définitivement du processus de division cellulaire [85].

La division cellulaire a été observée pour la première fois en 1842 par le botaniste Karl Nägeli lors de la formation du pollen [114]. Quarante ans plus tard, Walther Flemming perce le secret et découvre que l'ADN compacté se transforme en filament lorsque la cellule se divise. Cette observation a été rendue possible en suivant la coloration de la structure compactée de l'ADN qu'il nomme **chromatine** [50]. Durant l'interphase, la chromatine se compacte davantage et

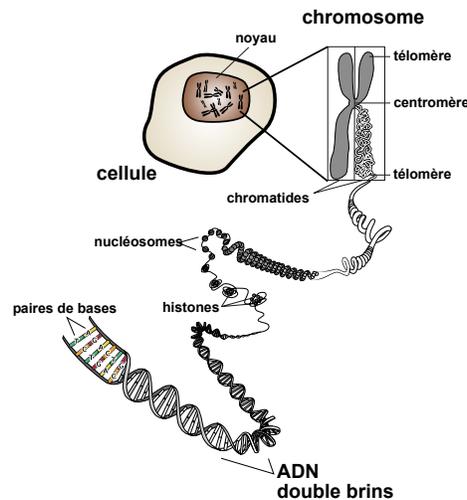


FIGURE 2.2 – Les différentes compactions de l’ADN. Dans le noyau, l’ADN s’enroule autour des protéines histones pour former les nucléosomes, unités de base de la chromatine. Cette chromatine se compacte pour former un chromosome. Dans cette figure, la molécule d’ADN a été dupliquée et le chromosome contient deux brins d’ADN fortement condensés appelés chromatides.

forme le **chromosome**, voir figure 2.2¹. La structure très condensée d’une molécule d’ADN d’un chromosome est appelée **chromatide**. L’ADN contenant 3.4×10^9 bases chez les mammifères est fortement compacté afin de protéger ce matériel génétique de toute dégradation et permet le déroulement de la mitose dans le but de répartir l’ADN de manière identique entre les cellules filles. Au cours de l’interphase, l’ADN est répliqué et les deux molécules d’ADN identiques restent attachées au niveau d’une zone de contact nommée **centromère** : les chromosomes contiennent alors deux chromatides.

Les phases du cycle cellulaire

Le cycle cellulaire est découpé en 4 phases. Trois phases sont observées au cours de l’interphase et une phase est nécessaire lors de la mitose. **L’interphase** entraîne l’accroissement du volume cellulaire et la duplication des chromosomes au cours de ses trois phases, voir figure 2.1 page ci-contre :

- ◇ la phase *G1* (de l’anglais *Growth 1*) assure la croissance et la préparation à la réplication de l’ADN,
- ◇ la phase *S* (de l’anglais *Synthesis*) réplique l’ADN,
- ◇ la phase *G2* (de l’anglais *Growth 2*) poursuit la réplication de l’ADN si besoin, et maintient la croissance de la cellule pour préparer sa division. De plus, le **centrosome** consti-

1. Image sous Licence Creative Commons placée dans le domaine du public.

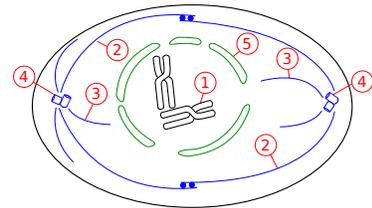
Chapitre 2 : Deux cycles régulateurs chez les mammifères et leurs modélisations

tuant le centre organisationnel des fibres de cytosquelettes ou **microtubules** est dupliqué. Les deux centrosomes s'éloignent l'un de l'autre jusqu'à atteindre des pôles opposés dans la cellule.

En réalité, il existe une autre phase pour laquelle la cellule est dans un état physiologique qui ne lui permet pas de se diviser : c'est la phase de **quiescence** ou phase G_0 . Plutôt que démarrer la phase G_1 , la ou les nouvelles cellules formées peuvent rentrer en quiescence et y rester quelques heures, quelques années, ou indéfiniment comme les cellules neuronales. A l'inverse, similairement aux cellules épithéliales, de nombreuses cellules ne sont jamais dans un état de repos. Finalement, les cellules quiescentes pouvant se diviser ultérieurement retournent dans un état physiologique qu'on observe durant la phase G_1 . Récemment, des études ont montré qu'il existe deux phases de quiescence qui dépendent de l'activité de la protéine Cdk2 [121].

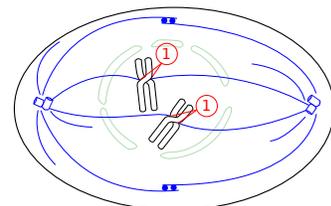
La seconde période du cycle cellulaire appelée **phase mitotique** M ou mitose assure la division de la cellule en deux cellules identiques ayant les caractéristiques de la cellule parente. Cette phase se divise en six étapes [119, 2] :

- ◇ la **prophase** permet la condensation de la chromatine en chromosome (1). Cette étape assure également la formation d'une variété de microtubules (2) appelée **fuseaux mitotiques** (3) qui sont attachés au niveau d'un centrosome (4). Les fuseaux mitotiques sont nécessaires à la migration des chromatides sœurs vers les pôles opposés de la cellule durant sa division.

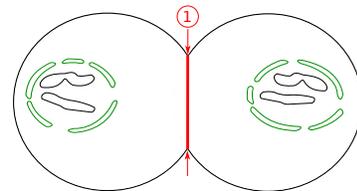
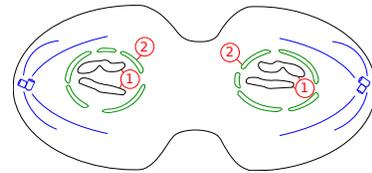
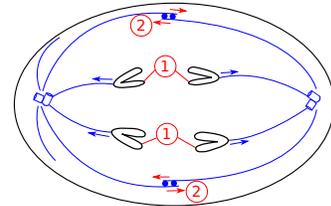
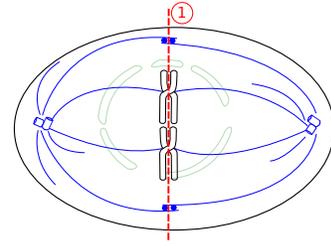


Enfin l'enveloppe nucléaire est fragmentée (5) suite à la phosphorylation (ajout d'un groupement phosphate) des protéines lamines. Ces protéines deviennent solubles et entraîne la déphosphorylation du maillage fibrillaire nommé lamina qui entoure le noyau. L'enveloppe nucléaire forme alors de nombreuses vésicules (5) car elle n'est plus soutenue par son maillage.

- ◇ la **prométaphase** fixe les chromosomes aux fuseaux mitotiques (1). Ces microtubules s'accrochent sur un ensemble de protéines appelé **kinétochores** situés au niveau du centromère du chromosome.



- ◇ la **métaphase** aligne les chromosomes au niveau du plan équatorial (1) de la cellule pour former la plaque métaphasique ou équatoriale. Cette étape vérifie que tous les chromosomes sont attachés à deux fuseaux mitotiques.
- ◇ l'**anaphase** sépare les chromatides sœurs (1) et les déplace vers les pôles opposés de la cellule. Les microtubules non attachés à des centromères appelés **microtubules polaires** (2) s'allongent entraînant l'éloignement des 2 pôles, ce qui permet d'éloigner les fuseaux mitotiques fixés au préalable sur un même chromosome. Ainsi, les chromatides sœurs s'éloignent l'une de l'autre.
- ◇ la **télophase** termine la division cellulaire. Les chromatides commencent à se décondenser (1), et les deux enveloppes nucléaires se reforment (2) suite à la déphosphorylation des protéines lamines, entraînant la reconstruction du maillage entourant l'enveloppe nucléaire.
- ◇ la **cytodiérèse** partage le cytoplasme de la cellule en deux. Un anneau composé de protéines d'actine et de myosine de type II (1) apparaît au niveau du plan équatorial. Cet anneau se contracte et entraîne une invagination de la cellule qu'on appelle **sillon de division**. Finalement, l'anneau se referme et permet la formation de deux nouvelles cellules.



Bien que l'interphase et la mitose constituent les éléments principaux du cycle cellulaire, il existe deux autres mécanismes permettant à une cellule de croître sans se diviser : ce sont les phénomènes d'**endomitose** et d'**endoréplication** décrits en 1939 par Lothar Geitler [52, 53]. Lors de l'endomitose, la cellule déclenche des processus de la phase mitotique, puis arrête instantanément la mitose pour démarrer un nouveau cycle cellulaire en début ou milieu de phase $G1$: la cellule a augmenté son volume, a dupliqué son ADN et ne s'est pas divisée. L'endoréplication présente un mécanisme similaire sur le cycle cellulaire sans effectuer de processus mitotique. Par exemple, la thrombopoïèse représente l'ensemble des processus de production de plaquettes sanguines et nécessite de multiples endoréplication de la cellule. Les cellules de

Chapitre 2 : Deux cycles régulateurs chez les mammifères et leurs modélisations

la moelle osseuse nommées mégacaryoblastes augmentent leur volume (30 microns à 150 microns) et forment les mégacaryocytes contenant, après 7 endoréplifications, 128 fois la quantité d'ADN normale. Ces cellules se fragmenteront par la suite en milliers de plaquettes sanguines et aideront au processus de coagulation du sang et à la cicatrisation de la plaie.

Mécanismes de surveillance

Les processus biologiques décrits précédemment sont essentiels pour le développement cellulaire et à la survie de nombreux organismes. Le bon déroulement des différentes phases est capital pour transmettre le matériel génétique intact à la descendance. Ainsi, le processus de division cellulaire requiert plusieurs **points de contrôles** ou *checkpoints* pour s'assurer de la correcte réplification de l'ADN et aussi de la bonne répartition du matériel génétique entre les cellules filles. Les mécanismes de surveillance ont été découverts par Leland Hartwell [61] et sont les suivants :

- ◊ le point de contrôle au niveau de la transition *G1/S* appelé **DDCP** (DNA Damage Checkpoint) vérifie que l'ADN ne soit pas endommagé avant de procéder à sa réplification,
- ◊ le point de contrôle au niveau de la transition *G2/M* appelé **RCP** (Replication Checkpoint) poursuit la vérification de l'endommagement de l'ADN, vérifie sa complète et unique réplification par cycle,
- ◊ le mécanisme de surveillance mitotique **MPC** (Mitotic Checkpoint) appelé aussi **SAC** (Spindle Assembly Checkpoint) a lieu entre les étapes de métaphase et d'anaphase. Il consiste à stopper la mitose en métaphase si des kinétochores ne sont pas attachés à des fuseaux mitotiques.

Dans les trois cas, le cycle cellulaire s'arrête tant que la réparation n'a pas lieu. Si la réparation est impossible, la cellule entre en **apoptose**, c'est-à-dire que la cellule entre dans le processus programmé d'auto-destruction en réponse à un signal.

Approche mécanistique

Les molécules clés du cycle cellulaire. La découverte du mécanisme de contrôle du cycle cellulaire est attribuée à trois éminents scientifiques que sont Leland Hartwell, Timothy Hunt et Paul Nurse, lauréats du prix Nobel de physiologie/médecine en 2001. Ils ont notamment identifié les protéines fondamentales gouvernant le cycle cellulaire et ont montré un fonctionnement identique chez la levure, les plantes ou l'Homme. En effet, les processus biologiques des phases du cycle cellulaire sont sous la direction de trois familles de protéines : les Cdks, les Cyclines et les CKI.

Les protéines oscillantes **Cdks** (Cyclin-Dependant Kinases) du cycle cellulaire ont été dé-

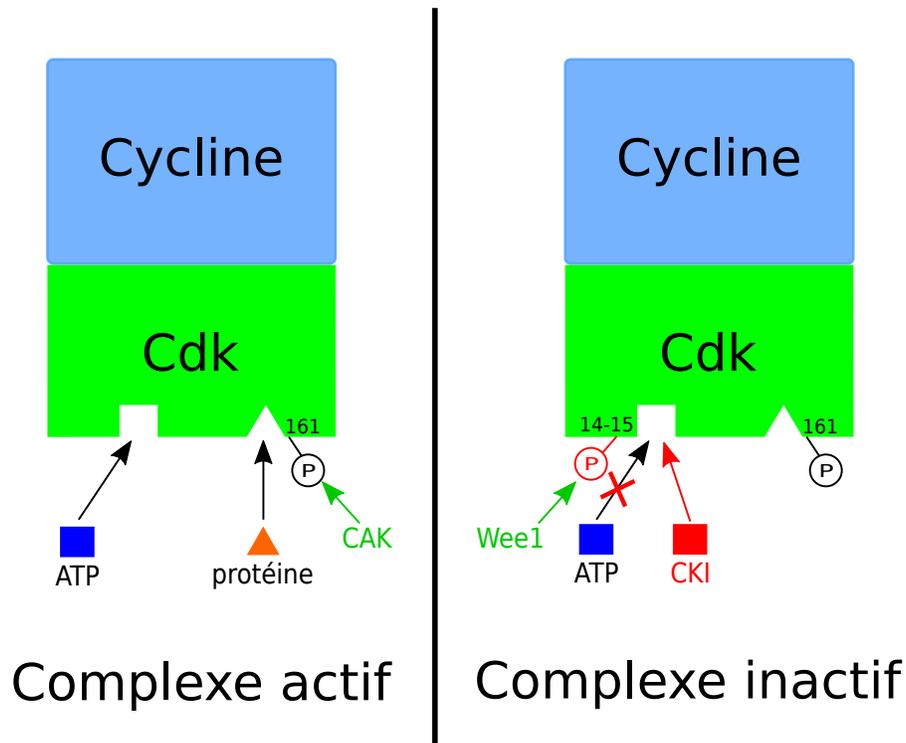


FIGURE 2.3 – Comportements actifs et inactifs des différents complexes Cycline/Cdk. Lorsque seul le site Thréonine 161 du Cdk est phosphorylé, le complexe protéique Cyc/Cdk est actif et permet d'ajouter un ion phosphate sur une protéine cible. Lorsque les sites 14 et 15 sont phosphorylés, le complexe protéique est inactif.

couvertes par Paul Nurse [101] et permettent de catalyser les réactions de phosphorylation, c'est-à-dire qu'elles favorisent le transfert d'ion phosphate de l'ATP vers une protéine substrat. Pour être active, chaque Cdk requiert d'être associée à une cycline [26, 49] et que le site spécifique Thréonine 161 du Cdk (acide aminé en position 161) soit phosphorylé par le complexe CAK (Cdk Activating Kinase) [79], voir figure 2.3. A l'inverse, les Cdks sont inactives dès que la Thréonine 14 et la Tyrosine 15 sont phosphorylées, même si la Thréonine 161 est phosphorylée. Ces phosphorylations inhibitrices sont dues à la protéine inhibitrice **Wee1** entre autres.

Timothy Hunt a identifié en 1983 les protéines oscillantes **Cyclines** [44] qui se complexent avec les Cdks et qui sont dégradées au cours des différentes phases, notamment lors de la mitose.

Enfin, d'autres protéines inhibitrices plus communément appelées **CKIs** (Cyclin-dépendant Kinase Inhibitor) se fixent sur les sites d'action des Cdks et bloquent leurs activités. Les CKIs sont regroupées en deux familles : les membres de la famille Cip/Kip inhibent Cdk2 et les membres de la famille INK4 inhibent Cdk4 et Cdk6 [117, 135, 132].

Bien que ces trois familles participent au déroulement du cycle cellulaire, le cycle cellulaire n'est

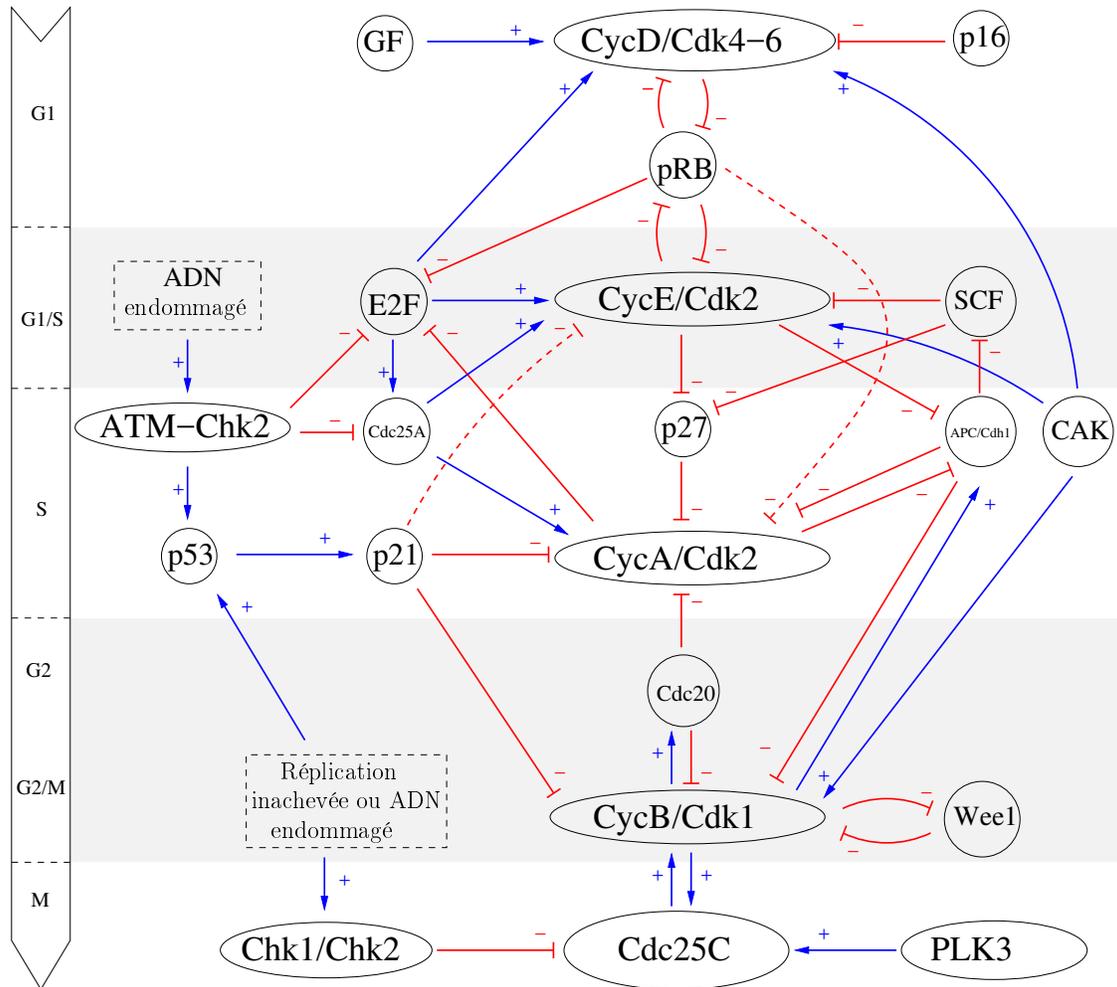
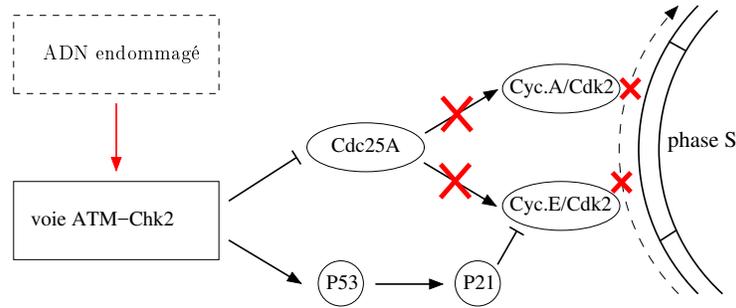


FIGURE 2.4 – Modèle mécanique complet du cycle cellulaire.

possible que sous certaines conditions. Ces conditions constituent des points de contrôle gouvernés par trois protéines : la protéine **p53** et les kinases **ATM** (Ataxia-Telangiectasia Mutated) et **ATR** (Ataxia-Telangiectasia Related). La protéine ATM s'active lorsque la molécule d'ADN est coupée ou lors de radiations ionisantes. Elle bloque le cycle cellulaire en activant plusieurs voies de signalisation et active notamment la protéine p53 [9, 28, 78]. De la même manière, ATR s'active lors d'erreur de réplication et arrête le cycle cellulaire.

Enfin, la protéine p53 est le facteur de transcription couramment appelé gardien du génome. Elle bloque la division cellulaire lors de lésion de l'ADN ou lors de stress cellulaire (oxydation, pH, pression osmotique, *etc*). Si la molécule d'ADN est réparée et les anomalies résolues, la division cellulaire reprend. Sinon, la cellule entre en apoptose.

La figure 2.4 décrit les mécanismes d'actions des différentes phases du cycle cellulaire. Ces

FIGURE 2.5 – Mécanisme de surveillance lors de la transition $G1/S$.

mécanismes d’actions peuvent paraître complexes au premier abord mais le point essentiel à retenir est **l’irréversibilité** du cycle cellulaire [100] : à chaque phase du cycle cellulaire, une molécule est stimulée pour assurer le rôle de la phase en cours tout en inhibant la phase précédente.

Mécanismes d’action de la phase $G1$ et de la transition $G1/S$.²

Le cycle cellulaire démarre en phase $G1$ en présence de **facteurs de croissance** GF (Growth Factor) lorsqu’ils sont en quantité suffisante. A partir des récepteurs intracellulaires de type tyrosine kinase et des récepteurs couplés à des protéines G, GF induit la voie MAP-Kinase [26] pour synthétiser les protéines **Cycline D** (1-3) (CycD), Cdk4 et Cdk6. Ces protéines forment un complexe qui phosphoryle son inhibiteur qui est la protéine rétinoblastome **pRB** [54] : pRB devient inactive. La protéine pRB ne séquestre plus le facteur de transcription E2F, permettant la poursuite du cycle cellulaire. En effet, E2F stimule la transcription de *cdc25A*, des *cyclines E* et *A* ce qui permet l’accumulation de leur protéine respective. La protéine Cdc25A formée va ensuite déphosphoryler les **cyclines E** (CycE) et **A** (CycA) pour permettre leurs activations [25]. La cycline E participe à la transition entre les phases $G1$ et S et la cycline A initie la phase S . La transition $G1/S$ démarre lorsque CycE est active. Cette cycline active la protéine kinase Cdk2, renforçant l’inhibition de pRB [113]. Finalement, le cycle cellulaire atteint un point de non-retour ou **point de restriction** signifiant la poursuite des phases jusqu’à la division cellulaire même si les facteurs de croissance diminuent.

Le point de contrôle $G1/S$ assure que l’ADN n’est pas endommagé, voir figure 2.5. Si c’est le cas, la voie de signalisation ATM-Chk2 est activée. A partir de cette voie de signalisation, deux mécanismes sont possibles. Le premier est une voie rapide d’arrêt du cycle cellulaire consistant à inhiber les facteurs de transcription E2F et *a fortiori* les *cyclines E* et *A* [137]. La seconde

2. Pour différencier les gènes des protéines, nous choisissons la nomenclature où les gènes sont écrits en italique et minuscule (*bm11*) et leurs protéines associées en lettres capitales (BMAL1) ou *a minima* la première lettre (Cdk2) (exceptées les protéines pRB, p21, p27 et p53).

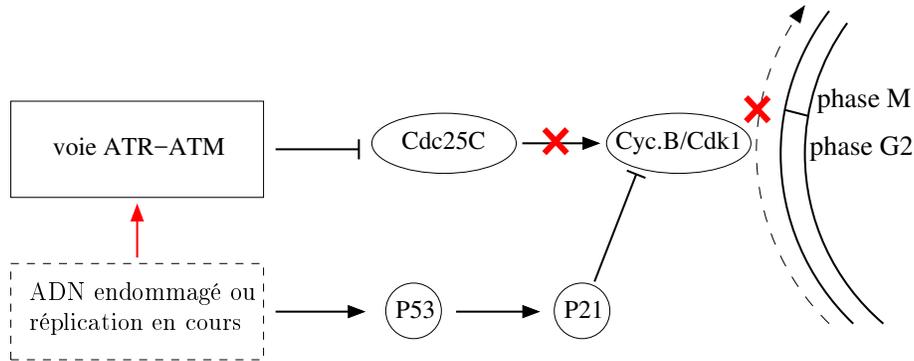


FIGURE 2.6 – Mécanisme de surveillance lors de la transition $G2/M$.

est une voie lente qui permet une accumulation de la protéine p53 ce qui active la CKI p21 et inhibe la *cycline E*.

Le point de contrôle agit au cours de la phase $G1$ ($G1$ *arrest*) ou en phase S (S *delay*) et mène à l'apoptose si la correction du problème est impossible.

Mécanismes d'actions de la phase S . Lors de la phase S , la protéine Cycline A s'accumule. La protéine Cdk2 se complexe avec cette cycline et assure la phosphorylation des protéines impliquées dans la réplication de l'ADN [113]. La protéine Cadhérine E codée par le gène *cdh1* est inactivée lorsqu'elle est phosphorylée par le complexe CycE/Cdk2. Cette inactivation permet l'accumulation du dimère CycA/Cdk2. De plus, CycA/Cdk2 renforce l'inactivité de *cdh1* assurant alors la synthèse de la **cycline B** (CycB) [54], et entraîne la dégradation de CycE par le complexe protéique SCF (SKP2/Cullin/F-box). Dès lors que la protéine Cycline B est synthétisée, la phase $G2$ démarre.

Mécanismes d'action de la phase $G2$ et de la transition $G2/M$. La phase $G2$ consiste à accumuler en abondance le complexe **CycB/Cdk1** dont la synthèse est activée indirectement par CycA/Cdk2. Au début de cette phase, le complexe APC/Cdh1 est inactivé par CycE/Cdk2 ce qui autorise l'accumulation des dimères CycA/Cdk2 [119]. CycB est alors synthétisée mais son complexe CycB/Cdk1 formé reste inactif car les résidus Thréonine 14 et Tyrosine 15 sont phosphorylés par la kinase Wee1 qui est aussi régulée par l'horloge circadienne [54]. Lorsque la réplication s'achève, la phosphatase Cdc25C passe dans le noyau et devient active dès que Plk1 la phosphoryle [126]. CycB/Cdk1 est alors déphosphorylé, puis le complexe phosphoryle son inhibiteur Wee1 pour le rendre inactif et son activateur Cdc25C pour renforcer son activation [79] : c'est un processus d'auto-activation. Plk1 phosphoryle également Wee1 pour l'inhiber.

Enfin, le point de contrôle $G2/M$ vérifie que l'ADN n'est pas endommagé et qu'il a été

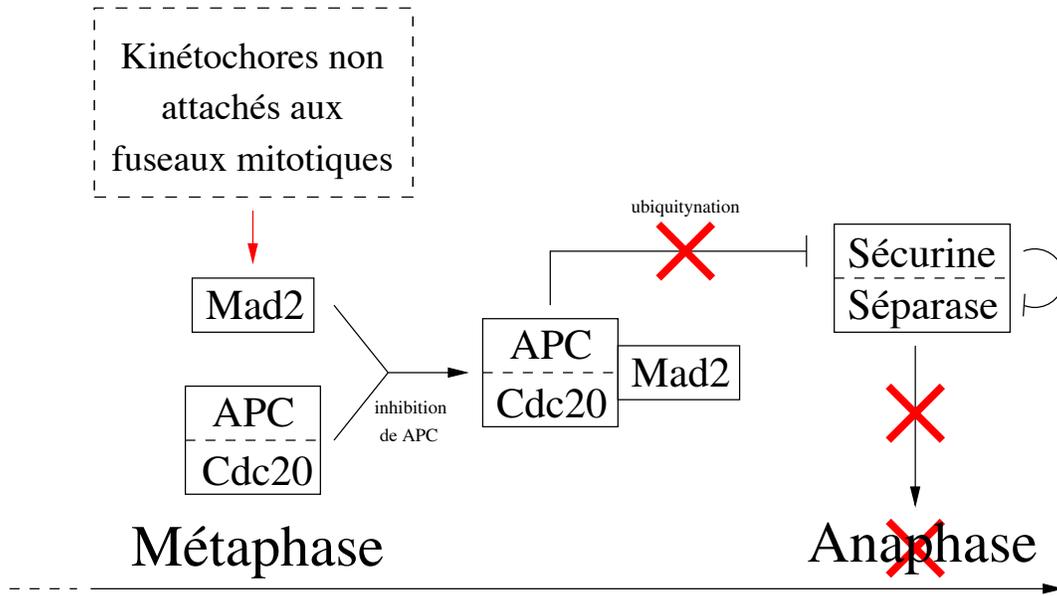


FIGURE 2.7 – Mécanisme de surveillance lors de la mitose.

entièrement répliqué, voir figure 2.6 page ci-contre. Pour réaliser la réparation ou terminer la réplication de l'ADN, la voie ATR/ATM Chk1-Chk2 inhibe Cdc25C en le phosphorylant ce qui le maintient dans le cytoplasme : la protéine Cdc25C ne peut plus activer le complexe CycB/Cdk1. De plus, l'endommagement de l'ADN active la voie de signalisation contenant p53 en accumulant cette protéine, ce qui active p21 et bloque alors l'activité de CycB/Cdk1.

Mécanismes d'actions de la phase M. Rappelons que pour former deux cellules filles avec un même matériel génétique, la mitose doit séparer les chromatides sœurs des chromosomes lors de l'anaphase.

Au début de la mitose, le complexe CycB/Cdk1 phosphoryle le complexe enzymatique **APC** (Anaphase Promoting Complex) et permet d'augmenter son affinité à Cdc20. Ce nouveau complexe entraîne l'ubiquitynation de la protéine **sécurine** par les enzymes du protéasome [105], c'est-à-dire l'ajout de la molécule ubiquitine sur l'extrémité de la protéine cible pour la détruire, voir figure 2.7. La securine se détache de la protéine **séparase** qui devient active et peut couper les liaisons peptidiques (protéolyse) du complexe protéique **cohésine**. Puisque le complexe **cohésine** est responsable du maintien des chromatides sœurs, sa dégradation en fin de métaphase permet de détacher les chromatides sœurs et de poursuivre la formation des deux cellules filles.

Durant les premières étapes de la mitose, le mécanisme de surveillance peut s'activer si au moins un ensemble de protéines kinétochores n'est pas attaché à un fuseau mitotique. Les kinétochores non attachés activent la protéine Mad2 (Mitotic Arrest Deficient 2), voir figure 2.7

qui se fixe au complexe APC/Cdc20 pour l'inhiber. Il devient alors impossible d'inhiber la sécurine qui reste fixée à la séparase. Ainsi, la séparation des chromatides est impossible dès lors qu'il existe un kinétochore non attaché à un microtubule.

Enfin, le dernier mécanisme essentiel de la phase mitotique est la dégradation protéique de la Cycline B par le complexe APC/Cdh1 [119]. En d'autres termes, il existe un rétrocontrôle négatif entre CycB et APC.

Ainsi, les mécanismes de régulation du cycle cellulaire peuvent être résumés comme suit : les Cdks interviennent séquentiellement au cours des différentes phases du cycle. Le complexe Cyc/Cdk active la phase suivante et est inhibée par cette dernière. Cdk1 est essentielle au cycle cellulaire et peut prendre le rôle des autres Cdks si elles sont inactives et permet de conduire seule la division cellulaire [111]. Enfin, les points de contrôle des transitions *G1/S*, *G2/M* et métaphase/anaphase permettent d'assurer l'intégrité du matériel génétique par les mécanismes contrôlés par les kinases ATR, ATM et le complexe APC.

2.1.2 Cycle circadien

Caractéristiques générales

Chez les mammifères, de très nombreux rythmes biologiques ont une périodicité circadienne. Ce phénomène traduit la coordination temporelle des processus biologiques pour adapter l'organisme aux changements circadiens de l'environnement. Les caractéristiques générales des **rythmes circadiens** sont les suivantes :

- ◇ les rythmes circadiens permettent à l'organisme de maintenir son bon fonctionnement malgré les fluctuations circadiennes de l'environnement [1]. Ces rythmes, comme par exemple le sommeil, la régulation de la température corporelle, la concentration des protéines dans la cellule ou encore la sécrétion hormonale ont une oscillation d'environ 24 heures. Ils persistent en absence de repère temporel, ce qui démontre leur nature endogène,
- ◇ le mécanisme sous-jacent fonctionne à l'échelle cellulaire et implique des gènes horloges [145] tels que *per*, *bmal1*, *etc*,
- ◇ le rythme circadien se synchronise avec le cycle jour/nuit de 24 heures par les éléments externes ou environnementaux appelés **zeitgebers** (donneurs de temps tels que la lumière, la température, les activités, l'alimentation, *etc*),
- ◇ l'organisation est hiérarchisée entre l'**horloge centrale** située au niveau des noyaux suprachiasmatiques (NSC) de l'hypothalamus et les **horloges périphériques** qui sont régulées par l'horloge centrale pour les resynchroniser régulièrement. L'ensemble de ces horloges forme ce que l'on nomme le **système de coordination circadienne**. En

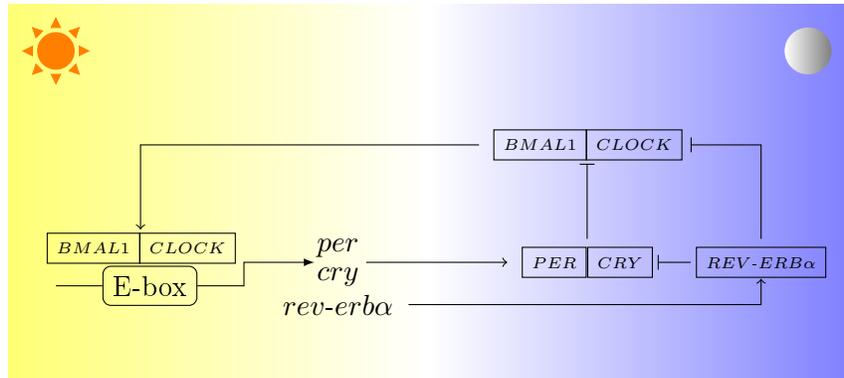


FIGURE 2.8 – Mécanismes d’action de l’horloge circadienne. Les activations sont représentées par des flèches \rightarrow tandis que les inhibitions sont représentées par \dashv . Le complexe *BMAL1/CLOCK* permet la synthèse des protéines du complexe *PER/CRY* et de *REV-ERB α* tandis que ces derniers inhibent *BMAL1/CLOCK* durant la nuit.

l’absence de synchronisation par l’horloge centrale, les horloges périphériques peuvent présenter au bout d’un certain temps un décalage de phase [144].

La lumière bleue participe au contrôle de l’horloge centrale [29, 95]. La lumière est détectée au niveau de l’œil par des photorécepteurs appelés cellules ganglionnaires, entraînant une stimulation des NSC *via* l’axe rétino-hypothalamique. Cette stimulation permet l’activation des gènes de l’horloge. Ces gènes de l’horloge oscillent et sont synchronisés avec la lumière bleue qui oscille selon le rythme nyctéméral (cycle jour/nuit de 24 heures).

Approche mécanistique

A l’instar du cycle cellulaire, les principaux mécanismes moléculaires de l’horloge circadienne ont été découverts par Jeffrey Hall, Michael Rosbash et Michael Young, lauréats du prix Nobel de physiologie/médecine en 2017. Ils ont identifié les gènes et protéines clés du rythme circadien appelés **cry** [42] et **timeless** [96] (*timeless* a été identifiée chez la drosophile). Les autres molécules essentielles du rythme circadien [123] sont **bmal1** [68, 72], **clock** [138] et **per** [81].

Les gènes de l’horloge contiennent une séquence d’ADN nommée **région E-box** dans la zone appelée **promoteur** qui contrôle la transcription de ces gènes. La région E-box participe à l’activation rythmique des gènes de l’horloge.

Durant la journée, pour un animal nocturne, le complexe protéique *BMAL1/CLOCK* se fixe sur des éléments de réponse E-box des gènes *per* et *cry* qui sont transcrits en ARN en fin de journée [71], voir figure 2.8.

Le complexe *BMAL1/CLOCK* se fixe également sur la région E-box du gène *rev-Erb α* et permet de synthétiser sa protéine au cours de la journée. Ensuite, la protéine *REV-ERB α*

inhibe la transcription du gène *bmal1* [30] : le rétrocontrôle négatif obtenu permet l'oscillation de BMAL1/CLOCK.

Au début de la nuit, les protéines PER et CRY sont synthétisées, puis forment un complexe qui inhibe son complexe activateur BMAL1/CLOCK. Ce second rétrocontrôle négatif assure alors l'oscillation de PER et CRY. A la fin de la nuit, les caséines kinases ϵ et δ phosphorylent PER et CRY et entraînent leurs dégradations par ubiquitynation [82, 140, 70]. Toujours présent durant la nuit, REV-ERB α inhibe la transcription de *cry* jusqu'à ce que CRY soit complètement dégradée [112, 140].

Finalement, les protéines PER et CRY sont totalement dégradées à l'aube. Les gènes *bmal1* et *clock* deviennent alors actifs et permettent la synthèse de leurs protéines respectives. Un nouveau cycle peut alors démarrer.

Au delà de ces mécanismes moléculaires qui sous-tendent l'horloge circadienne, les protéines clés du cycle circadien agissent sur l'activité de protéines externes à l'horloge. Le cycle circadien permet alors de contrôler d'autres systèmes biologiques. C'est notamment le cas de la protéine **Wee1**, dont le gène est transcrit suite à la fixation de BMAL1/CLOCK sur la région E-box du promoteur de *wee1* [92]. Rappelons que la protéine Wee1 inhibe la Cycline B au cours du cycle cellulaire [145, 54, 30]. Cette protéine est un premier lien entre les 2 cycles.

Dans les mécanismes du cycle cellulaire et du cycle circadien, nous avons vu que les rétrocontrôles négatifs et positifs ont un rôle essentiel pour assurer le bon fonctionnement du cycle. Ces structures permettent le bon déroulement des processus d'une part, et de coordonner les processus dans le temps en les produisant uniquement au bon moment d'autre part. Le mécanisme de rétrocontrôle est à la base de nombreux systèmes biologiques et ne peut être écarté dans la modélisation du phénomène étudié. Les rétrocontrôles seront donc incorporés dans le couplage du cycle cellulaire et du cycle circadien que nous souhaitons modéliser. Les sections suivantes expliquent d'abord les cadres de modélisation puis l'approche pour modéliser ce couplage.

2.2 Les cadres de modélisation

Caractéristiques générales

Dans ces travaux, la **modélisation** peut être vue comme la représentation d'un phénomène biologique complexe consistant à valider ou réfuter des hypothèses biologiques formulées.

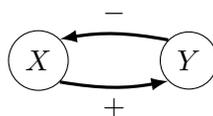


FIGURE 2.9 – Exemple de graphe d'influence. Les nœuds ou sommets sont représentés par des cercles étiquetés et les arcs ou arêtes sont des flèches éventuellement étiquetées. Le signe + représente une activation et le signe – représente une inhibition.

Chaque cadre de modélisation comprend deux éléments fondamentaux que sont le graphe d'influence entre les entités du système étudié et la manière dont on représente l'évolution des entités.

Un **graphe d'influence** d'un système biologique complexe représente d'une manière statique les différentes réactions moléculaires de ce système. Les entités biologiques du graphe sont représentées par des **nœuds** ou sommets et les influences entre les entités sont représentées par des flèches appelées **arcs** qui peuvent être étiquetées, voir figure 2.9. Les graphes d'influence sont donc statiques et ne représentent pas l'évolution des entités au cours du temps. Le cadre de modélisation complète la description statique du graphe d'influence et permet de suivre la trajectoire de chacune des entités du graphe selon différents niveaux de description [74].

Le **cadre de modélisation** définit alors l'univers des dynamiques que l'on peut associer à un graphe d'influence. Il contient l'ensemble des états possibles du modèle pour représenter l'évolution de ses entités. Les **paramètres dynamiques** gouvernant l'évolution de ces entités doivent être identifiés pour représenter complètement le comportement du système.

Les différents cadres de modélisation se distinguent par leurs niveaux de description plus ou moins détaillés (abstrait/concret) et se basent sur des connaissances différentes du système biologique. Il est possible de représenter très précisément un système biologique en décrivant l'évolution précise des concentrations de toutes les entités, comme il est possible d'abstraire ces comportements très précis en ne gardant que les activités (actif/inactif) de chacune des entités.

Les réseaux de régulation des systèmes biologiques peuvent être représentés en informatique par des graphes d'influence. A ces graphes est souvent associée une sémantique décrivant par exemple l'évolution du système biologique. Cependant, sans même connaître précisément le comportement d'un modèle, il est possible d'obtenir des informations sur sa dynamique : c'est ce qu'on appelle l'**analyse statique**. L'analyse statique englobe différentes méthodes formelles, l'une d'elle par **interprétation abstraite** permet d'approximer le comportement du système [38] et a montré son intérêt dans des modèles pour différents niveaux de détails. En effet, si un comportement n'est pas observé dans la sémantique (la signification des termes) d'un cadre de modélisation avec un niveau abstrait maximal, alors il est impossible d'observer

ce même comportement dans toutes les sémantiques moins abstraites [74].

Pour choisir le cadre de modélisation adapté au système biologique étudié, il est utile de se poser les questions suivantes :

- ◇ Les niveaux quantitatifs des différents composants sont-ils connus ?
- ◇ Connait-on la dynamique des composants en temps continu ?
- ◇ Y a-t-il une grande variabilité des résultats expérimentaux ?

Les réponses à ces questions aident à choisir parmi les différents niveaux de description et donc le cadre de modélisation, le goulot d'étranglement étant d'identifier les paramètres dynamiques gouvernant le système biologique.

Niveaux d'abstraction des cadres de modélisation

On peut classer les différents cadres de modélisation en 4 catégories.

Les **cadres de modélisation différentiels** se basent sur une sémantique continue où les concentrations des molécules peuvent être mesurées expérimentalement au cours du temps. L'évolution des différentes entités est décrite par des équations différentielles ordinaires (ODE) de la forme $\frac{dx}{dt} = f(x)$. Le cadre de modélisation différentiel est utile lorsqu'on cherche à représenter quantitativement chaque composant biologique du graphe d'influence sans avoir observé d'effets stochastiques.

Les **cadres de modélisation stochastiques** présentent un niveau d'interprétation concret et s'intéresse à représenter l'évolution en temps continu du nombre de molécules de chacune des espèces étudiées tout en considérant que les événements faisant passer d'un état à un autre ont une nature stochastique. Un exemple classique est le processus de Markov à temps continu où le temps passé dans les états du système est une valeur aléatoire positive qui suit une loi exponentielle. Les cadres de modélisation basés sur cette approche s'apparentent la plupart du temps à des modèles différentiels en y incluant la variabilité des résultats expérimentaux.

Les **cadres de modélisation qualitatifs** ou discrets se basent sur une interprétation plus abstraite des observations. Les entités biologiques sont représentées par leur présence ou leur absence dans les états du système. C'est le cas de la sémantique des réseaux de Petri [62, 31] ou encore la sémantique booléenne avec des systèmes de transitions asynchrones [76]. Selon la sémantique utilisée, les données quantitatives n'ont pas à être connues précisément (sémantique booléenne), ainsi que le temps passé dans chacun des états qualitatifs (sémantique des réseaux de Petri).

Enfin, les **cadres de modélisation hybrides** mixtent différents niveaux d'abstraction de modèles. Les approches hybrides reliant expression booléenne des gènes et niveau quantitatif continu des protéines sont un exemple de cadre hybride, comme les modèles continus

stochastiques [47]. Puisque la biologie nécessite de combiner plusieurs points de vues, il n'est pas étonnant de remarquer que de nombreux recouvrements entre les cadres de modélisation existent [12] :

- ◊ l'approche PLDE³ (Piecewise Linear Differential Equation) approxime les équations différentielles ordinaires contenant des fonctions de Hill (équations décrivant la coopération des composants sous forme de fonction sigmoïde) par des fonctions "steps" et ainsi représente l'évolution continue des composants dans chaque état du système [57, 74],
- ◊ les réseaux de Petri hybrides fonctionnels (HFPN pour Hybrid Functional Petri Net) sont des approches qualitatives avec deux caractéristiques (parties discrètes et parties hybrides) où les transitions entre les états du système sont associées à une fonction dépendant des taux de réaction chimique [3, 91]. Par exemple, le contrôle génétique (représenté par des parties discrètes) est relié aux réactions métaboliques (représentés par des parties continues),
- ◊ les réseaux de Petri stochastiques présentent des transitions entre les places et le temps pour les franchir est non déterministe, représenté par une variable aléatoire positive. Par exemple, les valeurs de ces variables peuvent représenter des constantes cinétiques pour des réactions entre composants.

Cependant, plusieurs modèles ODEs mènent à un unique modèle discret alors que ces modèles continus décrivent des informations différentes. Ainsi, l'interprétation abstraite serait utile pour unifier les différents cadres de modélisation [74].

2.3 Modèles existants

Pour modéliser le couplage entre le cycle cellulaire et le cycle circadien, nous devons d'abord construire son graphe d'influence et choisir le cadre de modélisation à utiliser. Pour nous aider dans cette tâche, les modèles existants du cycle cellulaire, du cycle circadien et du couplage de ces deux cycles sont présentés dans cette section.

2.3.1 Les modèles du cycle cellulaire

Les modèles du cycle cellulaire existants se différencient selon le type cellulaire étudié, le cadre de modélisation choisi et le nombre d'entités du graphe d'influence. Une liste non exhaustive de modèles du cycle cellulaire est présentée en Table 2.1 page suivante complétant la liste présentée en [48]. Le nombre d'entités entre les graphes varie selon le niveau de détail souhaité. La régression linéaire pouvant expliquer le nombre de paramètres en fonction du

3. PLDE est aussi appelé PADE (Piecewise Affine Differential Equation)

n°	Références	Cadre de modélisation	Organisme	Entités	Paramètres
1	[133] ⁴	Différentiel	Mammifère, levure	4	-
2	[142]	Différentiel	Mammifère	7	13
3	[46]	Qualitatif	Mammifère	10	10
4	[75]	Différentiel	Algue unicellulaire	10	32
5	[56]	Différentiel	Mammifère	10	39
6	[88]	Qualitatif	Levure	11	11
7	[119]	Hybride	Mammifère	11	19
8	[108]	Différentiel	Eucaryote	14	31
9	[21]	Différentiel	Levure	28	40
10	[10]	Stochastique	Levure	34	157
11	[106]	Différentiel	Mammifère	44	203

TABLE 2.1 – Exemples de modèles du cycle cellulaire. Les colonnes *Entités* et *Paramètres* indiquent le nombre d’entités présentes dans le graphe d’influence et le nombre de paramètres à identifier, respectivement.

nombre d’entités est : $\#paramètres = 4.88 \times \#entités - 31.87$ avec un coefficient de corrélation R^2 de 0.849. Ce résultat indique que le nombre de paramètres augmente lorsque le nombre d’entités augmente.

Dans ces exemples, les modèles du cycle cellulaire varient selon les éléments mis en avant. En effet, les modèles peuvent se focaliser sur l’activité des protéines kinases, sur les transitions du cycle cellulaire, ou encore sur les concentrations des protéines telles que les complexes Cycline/Cdk. Ces différents modèles ont confirmé des hypothèses formulées au départ. Par exemple, le modèle différentiel $n^\circ 8$ chez les eucaryotes [108] a montré que les transitions *G1/S* et *G2/M* ne peuvent être franchies que si les mécanismes d’action ont assuré que l’ADN n’est pas endommagé et est correctement répliqué. Il est alors possible de poursuivre le cycle cellulaire en phase *S* ou *M* selon la transition. De plus, il a été prouvé dans ce modèle que l’activité des complexes Cyclines/Cdks dépend des protéines Cdc25 et Wee1 : Cdc25 active les complexes Cyclines/Cdks tandis que Wee1 les inhibe. Un couplage entre ce modèle et un modèle circadien a déjà été réalisé [127] et sera présenté en sous-section 2.3.3 page 39.

Dans un second exemple, le modèle hybride $n^\circ 7$ chez les mammifères [119] est intéressant car il permet de coupler des équations différentielles linéaires par morceau (modèle différentiel) avec des variables booléennes (modèle qualitatif). La synthèse et la dégradation des cyclines sont décrites précisément tandis que les niveaux d’activité des facteurs de transcription et des complexes permettant l’ubiquitination sont représentés par la valeur 0 (inactive) ou 1

4. Cet article décrit uniquement une abstraction du cycle cellulaire chez différentes espèces par un graphe d’influence ayant 4 entités. Il n’y a donc ni de dynamique associé ni de paramètres à identifier.

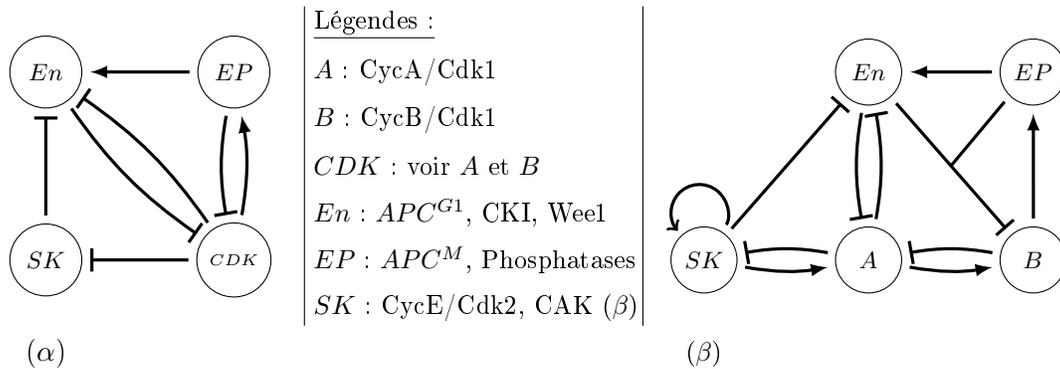


FIGURE 2.10 – 2 graphes d’influence du cycle cellulaire. (α) : le graphe introduit par Tyson [133]. (β) : notre graphe d’influence. L’entité SK (pour Starter Kinase) contient dans (α) le complexe $CycE/Cdk2$ et dans (β) les complexes $CycE/Cdk2$ et CAK .

(active). Les résultats obtenus sur l’abondance des cyclines sont en accord avec les observations expérimentales mais quelques incohérences existent nécessitant des modifications du modèle. Par exemple, la cycline A est synthétisée dès qu’elle atteint 8% de son niveau maximum, alors que leurs observations expérimentales montrent une synthèse dès 5% de son niveau maximum.

Dans nos travaux, la représentation du cycle cellulaire chez le mammifère s’inspire du graphe d’influence de Tyson et Novak [133] contenant 4 entités, voir figure 2.10. Le cycle cellulaire des mammifères et des levures est représenté par ce graphe. Chaque nœud du graphe regroupe des protéines ayant une même activité. Chez les mammifères, les entités contiennent les protéines suivantes :

- ◊ SK (*Starter Kinase*) représente le complexe $CycE/Cdk2$ essentiel à la fin de $G1$ et de la transition $G1/S$ ⁵,
- ◊ CDK regroupe les complexes protéiques $CycA/Cdk1$ et $CycB/Cdk1$ présents durant les phases S , $G2$ et M ,
- ◊ En (*Cdk Enemies*) contient les protéines inhibitrices en phase $G1$ et $G2$ telles que APC , $Wee1$ et les CKIs $p21$ et $p27$,
- ◊ EP (*Exit Proteins*) contient les protéines inhibitrices de la phase M (APC et les phosphatases).

Nous avons choisi de dissocier les complexes de l’entité CDK du graphe d’influence de Tyson en deux entités distinctes (A et B) car ils interviennent à différentes phases du cycle cellulaire. Les activations et inhibitions que pouvaient présenter les entités En , EP et SK sur CDK ont été modifiées en conséquence. De plus, l’entité SK est inhibée lorsque l’ADN est endommagé (non indiqué dans le graphe (α)). Pour représenter cette information, nous avons inclus le

5. Dans la figure 2.10, l’entité SK contient aussi le complexe CAK pour le graphe (β).

n°	Références	Cadre de modélisation	Organisme	Entités	Paramètres
1	[35]	Qualitatif	Mammifère	3	8
2	[130]	Différentiel	Souris, Neurospora	3	9
3	[14]	Hybride	Mammifère	4	20
4	[110]	Différentiel	Drosophile	6	16
5	[131]	Différentiel	Drosophile	10	35
6	[60]	Stochastique	Arabidopsis thaliana	10	90
7	[141]	Différentiel	Drosophile	19	47
8	[140]	Différentiel	Mammifère	21	100

TABLE 2.2 – Exemples de modèles du cycle circadien.

complexe CAK (Cdk Activating Kinase) dans l'entité SK [83] et nous avons ajouté une boucle de rétroaction positive sur SK qui agit lorsque le mécanisme de surveillance en *G1/S* montre que l'ADN n'est pas endommagé.

2.3.2 Les modèles du cycle circadien

Les premiers travaux de modélisation du rythme circadien ont été réalisés en 1966 par Arthur Winfree [139]. Depuis ces travaux, de nombreux autres modèles ont été réalisés notamment chez la cyanobactérie, montrant l'importance des gènes de l'horloge dans les boucles de rétroaction [73]. De la même manière que le cycle cellulaire, une liste non exhaustive de modèles du cycle circadien est présentée en Table 2.2.

La régression linéaire pouvant expliquer le nombre de paramètres en fonction du nombre d'entités est : $\#paramètres = 3.95 \times \#entités + 3.1$ avec un coefficient de corrélation R^2 de 0.598. Ainsi, le nombre de paramètres du graphe n'est que faiblement corrélé au nombre d'entités présentes.

Le cadre de modélisation choisi varie selon le niveau de détail fourni. Par exemple, le modèle différentiel $n^\circ 5$ chez la drosophile dont le graphe d'influence présente 10 entités a montré l'effet du cycle nyctéméral sur le cycle circadien [131]. Nous avons également montré l'influence de ce cycle jour/nuit à partir d'un graphe présentant 4 entités en utilisant un cadre de modélisation hybride [14]. Ce cadre de modélisation sera décrit dans le chapitre 3 page 43.

Pour réaliser le couplage entre le cycle cellulaire et le cycle circadien, nous avons choisi de ne retenir que 3 entités pour l'horloge circadienne. Ces 3 entités sont les complexes PER/-CRY, BMAL1/CLOCK et REV-ERB α , voir figure 2.11 page ci-contre. Cette figure représente d'un point de vue simplifié les influences de la figure 2.8 page 31. Le graphe d'influence présente une boucle de rétroaction négative entre BMAL1/CLOCK et PER/CRY. Pour rappel, BMAL1/CLOCK se fixe sur les promoteurs des gènes *per* et *cry* favorisant ainsi la synthèse

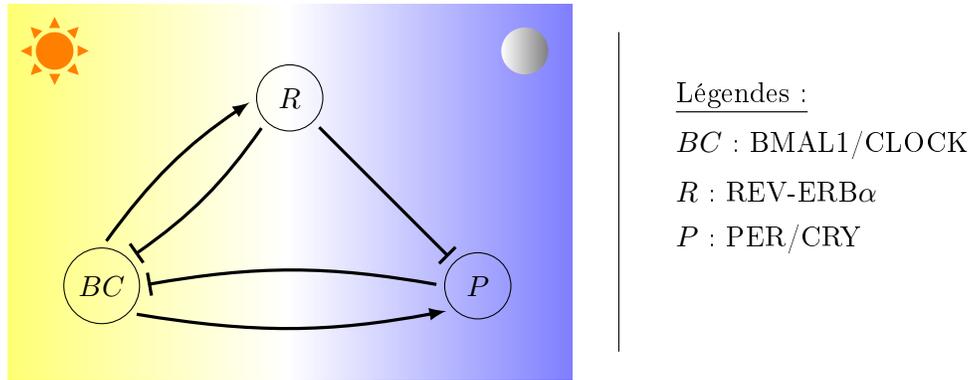


FIGURE 2.11 – Mécanismes d'action de l'horloge circadienne. Les activations sont représentées par la flèche \rightarrow tandis que les inhibitions sont représentées par l'autre flèche. Le complexe BMAL1/CLOCK permet la synthèse des protéines du complexe PER/CRY tandis que ce dernier inhibe BMAL1/CLOCK. BMAL1/CLOCK permet aussi de synthétiser la protéine REV-ERB α qui inhibe PER/CRY.

de leurs protéines, tandis que le complexe PER/CRY inhibe BMAL1/CLOCK. Dans ce graphe d'influence, il existe aussi une boucle de rétroaction négative entre BMAL1/CLOCK et REV-ERB α sur le même principe que la première rétroaction négative. Enfin, REV-ERB α inhibe le complexe PER/CRY.

2.3.3 Les modèles couplant les deux cycles biologiques

Des modèles du couplage cycle cellulaire et cycle circadien ont déjà été construits, voir Table 2.3 page suivante.

La régression linéaire pouvant expliquer le nombre de paramètres en fonction du nombre d'entités est : $\#\text{paramètres} = 2.99 \times \#\text{entités} + 0.52$, avec un coefficient de détermination R^2 de 0.994. Ce résultat indique que le nombre de paramètres augmente toujours quand le nombre d'entités du graphe augmente.

Le modèle $n^{\circ}6$ dont le graphe contient 34 entités combine le graphe d'influence du cycle cellulaire comprenant 14 entités chez les eucaryotes détaillé en sous-section 2.3.1 page 35 avec un graphe du cycle circadien représenté par 20 entités. Ce modèle a montré que la synthèse de *rev-Erb α* ou l'inhibition de la transcription de *bmal1* durant la mitose concorde avec les données expérimentales.

Un autre modèle remarquable est le modèle hybride $n^{\circ}1$ comprenant des équations différentielles stochastiques des phases de l'horloge et du cycle cellulaire [47]. L'idée est de coupler l'horloge circadienne et le cycle cellulaire compatibles avec les résultats expérimentaux obtenus. Les simulations ont montré des comportements oscillants coexistant entre le cycle cellulaire et

n°	Références	Cadre	Organisme	Entités	Paramètres	Direction
1	[47]	Stochastique	Mammifère	2	8	$H \rightarrow C$
2	[143]	Hybride	Cyanobactérie	2	8	$H \rightarrow C$
3	[22]	Stochastique	Mammifère	2	24	$H \leftrightarrow C$
4	[33]	Différentiel	Mammifère	5	10	$H \rightarrow C$
5	[41]	Hybride	Mammifère	10	33	$H \rightarrow C$
6	[127]	Différentiel	Mammifère	34	102	$H \leftrightarrow C$
7	[55]	Différentiel	Mammifères	39	187	$H \rightarrow C$

TABLE 2.3 – Exemples de modèles couplant le cycle cellulaire et le cycle circadien. La colonne *Direction* indique l’influence décrite entre l’horloge circadienne H et le cycle cellulaire C dans le graphe d’influence.

le cycle circadien.

2.4 Vers un nouveau modèle du couplage

Le couplage cycle cellulaire/cycle circadien peut être représenté à des niveaux de description très variés. Nous avons fait le choix d’abstraire les composants essentiels de chaque cycle par une même entité lorsqu’ils présentent un rôle identique sur leur cible commune. Notre graphe d’influence comporte 9 entités, avec 3 entités essentielles du cycle circadien (PER/CRY, BMAL1/CLOCK et REV-ERB α) et 5 entités pour le cycle cellulaire, voir figure 2.12. Nous avons aussi ajouté une entité supplémentaire qui représente la protéine $p53$ car elle influence d’une part le cycle cellulaire par les mécanismes de surveillance et d’autre part le cycle circadien par l’inhibition de *per2* [93]. Notons aussi que $p53$ est activée par phosphorylation grâce à CycB [23]. Cette phosphorylation empêche la dégradation de $p53$ et permet aussi de l’activer dans les mécanismes de surveillance si nécessaire.

Alors que plusieurs actions du cycle circadien sur le cycle cellulaire sont connues, celles du cycle cellulaire sur l’horloge circadienne restent peu étudiées. Les influences des entités entre ces deux cycles sont décrites en Table 2.4 page ci-contre. Par exemple, le cycle circadien agit sur le cycle cellulaire *via* l’activation du gardien du génome $p53$ par la protéine PER d’une part, et *via* les inhibiteurs des complexes Cycline/Cdk d’autre part. Ces influences mènent à bloquer ou à poursuivre le cycle cellulaire selon la protéine circadienne mise en jeu : BMAL1/CLOCK et PER bloquent la progression du cycle tandis que REV-ERB α la favorise. En effet, BMAL1/CLOCK se fixe sur la région E-box du promoteur *wee1* pour le synthétiser. Wee1 phosphoryle alors Cdk1

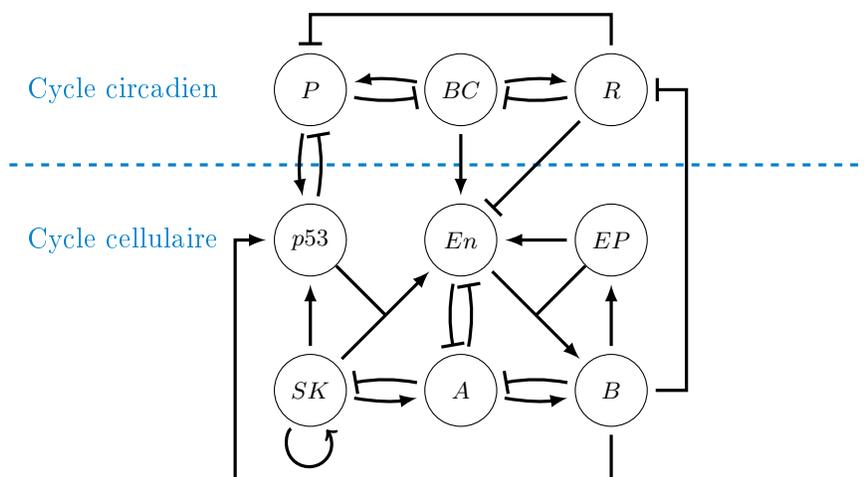


FIGURE 2.12 – Couplage du cycle circadien et du cycle cellulaire.

Sens	Influence	Signification	Référence
$C \rightarrow H$	$p53 \longrightarrow \perp P$	p53 inhibe PER2 en bloquant son promoteur	[93]
$C \rightarrow H$	$B \longrightarrow \perp R$	CycB/Cdk1 déstabilise REV-ERB α	[146]
$H \rightarrow C$	$BC \longrightarrow En$	BMAL1/CLOCK active la kinase Wee1	[92]
$H \rightarrow C$	$P \longrightarrow p53$	PER/CRY active p53 par complexation	[58]
$H \rightarrow C$	$R \longrightarrow \perp En$	REV-ERB α inhibe p21	[30, 59]

 TABLE 2.4 – Influences entre entités du cycle cellulaire et du cycle circadien. C représente le cycle cellulaire et H représente l’horloge circadienne.

complexée avec CycB entraînant ainsi son inhibition. A l’inverse, REV-ERB α inhibe les CKIs tels que p21, favorisant la poursuite du cycle cellulaire.

A l’opposé, l’action du cycle cellulaire sur l’horloge circadienne est peu connue. Les quelques informations connues à ce jour montrent que la protéine p53 bloque le promoteur de *per2* [93]. C’est une action indirecte sur le cycle circadien puisque p53 n’intervient pas spécifiquement dans le cycle cellulaire. Elle est cependant activée par CycB/Cdk1 et CycE/Cdk2. De plus, la kinase Cdk1, lorsqu’elle est complexée à CycB, semble phosphoryler REV-ERB α . Néanmoins, l’activité résultante de cette phosphorylation n’a pas été étudiée dans ces travaux. Cette influence est tout de même incluse dans le couplage afin d’étudier par modélisation son éventuel rôle. Toutefois, il est intéressant de noter que REV-ERB α est dégradée dans les cellules arrêtées en G2 avec du nocodazole [146].

En résumé, la modélisation de systèmes biologiques nécessite de définir d'une part un graphe d'influence entre les entités du système et d'autre part un cadre de modélisation pour décrire l'évolution des entités. Le graphe d'influence comprend des nœuds représentant un ou plusieurs composants biologiques et des arcs décrivant les activations et inhibitions entre les différentes entités du graphe. Il s'agit d'une synthèse statique des influences entre ces entités.

Après analyse des mécanismes intervenant dans le cycle cellulaire et dans le cycle circadien, nous avons défini un graphe d'influence du couplage dans l'optique d'analyser sa dynamique à partir d'observations expérimentales. Ce graphe comporte 3 entités intervenant dans le cycle circadien, 5 entités du cycle cellulaire et l'entité *p53* qui contrôle essentiellement le cycle cellulaire.

Nous avons vu que les cadres de modélisation définissent l'évolution des entités d'un graphe d'influence, gouvernée par les paramètres dynamiques. Le choix d'un cadre de modélisation n'est pas toujours aisé mais le niveau de description attendu ainsi que la disponibilité et la qualité des données sont des critères qui peuvent nous aider à choisir un cadre de modélisation adapté.

Chapitre 3

Définition du cadre de modélisation utilisé

3.1 Motivation

Les cadres de modélisations diffèrent selon les aspects qu'ils mettent en avant. Certains cadres se focalisent plus sur les aspects causaux (**modèles qualitatifs**), sur la variabilité des résultats expérimentaux (**modèles stochastiques**), sur la précision des informations données (**modèles différentiels**), *etc* [39]. Mais quelque soit le cadre choisi, la dynamique des modèles est contrôlée par des **paramètres dynamiques**, et la recherche des paramètres dynamiques d'un modèle représentant un système biologique donné reste une étape clef. L'idée maîtresse est alors d'utiliser la puissance de calcul de l'ordinateur pour aider le modélisateur à choisir ces paramètres dynamiques en utilisant les données disponibles.

Les **cadres de modélisation discrets** ou qualitatifs présentent l'avantage d'une identification quasi-automatique des paramètres dynamiques dès qu'on est capable d'écrire des propriétés sur les traces observées [36, 17]. Cependant, jusqu'à présent, ces cadres de modélisation abstraient totalement le temps qui sépare deux événements qualitatifs en ne retenant que l'ordre des événements, et ce même lorsque ces durées entre deux événements sont informatives et pourraient contraindre les paramètres dynamiques.

A l'opposé, les cadres de modélisation différentiels ou stochastiques décrivent de manière très précise l'évolution des différentes entités biologiques, mais l'identification des paramètres dynamiques est très compliquée [55] et l'automatisation de cette identification relève de la recherche. De plus, lorsqu'un jeu de paramètres est considéré valide, l'ajout d'entités supplémentaires dans le graphe d'influence du système biologique demande de recommencer totalement tout le processus d'identification des paramètres car l'ajout de régulation peut rendre caduque tout précédent

choix sur les paramètres.

Pour conserver les avantages de chaque cadre de modélisation, nous avons utilisé un **cadre de modélisation hybride** pour décrire l'évolution des entités du graphe d'influence. Ce cadre hybride conserve le côté qualitatif des modèles discrets en continuant de décrire les successions d'événements et permet de définir une évolution des entités biologiques entre ces différents événements.

Le cadre de modélisation hybride complète avantageusement le cadre discret :

- ◇ Les modèles discrets interprètent uniquement les observations correspondant aux actions d'une ou plusieurs entités sur une entité cible. Ces modèles ne représentent que qualitativement l'évolution au cours du temps des différentes entités.
- ◇ Les modèles hybrides permettent de prendre en compte le temps entre deux événements contrairement aux modèles discrets. L'évolution des entités entre les événements est graduelle.
- ◇ Les modèles hybrides peuvent inclure des comportements non représentés dans les modèles discrets tels que les phénomènes de saturation ou dégradation totale d'entités biologiques.

A l'opposé, dans le cadre différentiel ou stochastique, les algorithmes permettant l'identification des paramètres dynamiques du modèle nécessitent souvent un temps de résolution prohibitif (méthode des moindres carrés, recherche de sous-espace, *etc* [89]) et ne permettent pas de caractériser l'ensemble des jeux de paramètres compatibles avec les données d'observations. Et les observations biologiques expérimentales ne permettent généralement pas de reconstruire l'évolution continue des concentrations des entités biologiques. Ainsi, passer à un modèle hybride pourrait permettre d'automatiser au moins partiellement cette étape.

Les sections suivantes présentent le cadre de modélisation hybride que nous avons choisi pour décrire le comportement des graphes d'influence de systèmes biologiques. Nous expliquons au préalable le cadre de modélisation discret sur lequel s'appuie l'approche hybride.

3.2 Le cadre de modélisation de René Thomas avec multiplexes

Le cadre de modélisation discret de René Thomas défini en 1973 permet de représenter de manière qualitative la dynamique des entités d'un système biologique [124, 125]. Les modèles se basent sur des graphes d'influence où les entités sont des nœuds (l'ensemble des entités est noté V) et les influences sont représentées par des arcs orientés étiquetés, voir figure 3.1 page suivante. L'étiquette de l'arc représente le niveau d'expression que l'entité doit atteindre pour que l'arc soit actif, et son signe décrit le type d'activité : + pour une activation et - pour une inhibition.

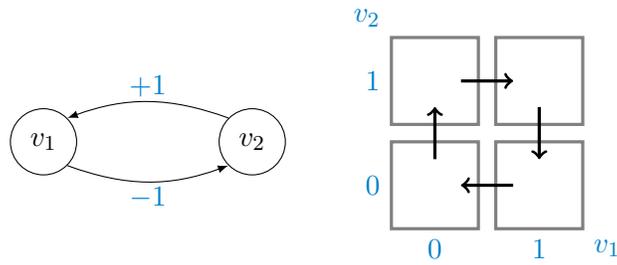


FIGURE 3.1 – Graphe d’influence représentant une boucle de rétroaction négative simple et son graphe d’états. Lorsque l’entité v_2 est active (niveau 1), elle active l’entité v_1 et la fait passer du niveau 0 au niveau 1. Une fois que v_1 a atteint le niveau 1, v_1 inhibe v_2 et la rend inactive.

Tout l’intérêt de l’approche discrète de René Thomas réside dans la représentation des dynamiques qui doivent être associées à ces graphes d’influence statiques. La méthode utilisée consiste à abstraire les concentrations quantitatives des entités biologiques par des niveaux qualitatifs. L’abstraction des concentrations est une simplification acceptable puisque les concentrations réelles ne sont pas précisément mesurables *in vivo*.

Cette abstraction repose sur l’hypothèse suivante : pour chaque influence $v_1 \rightarrow v_2$, le taux de

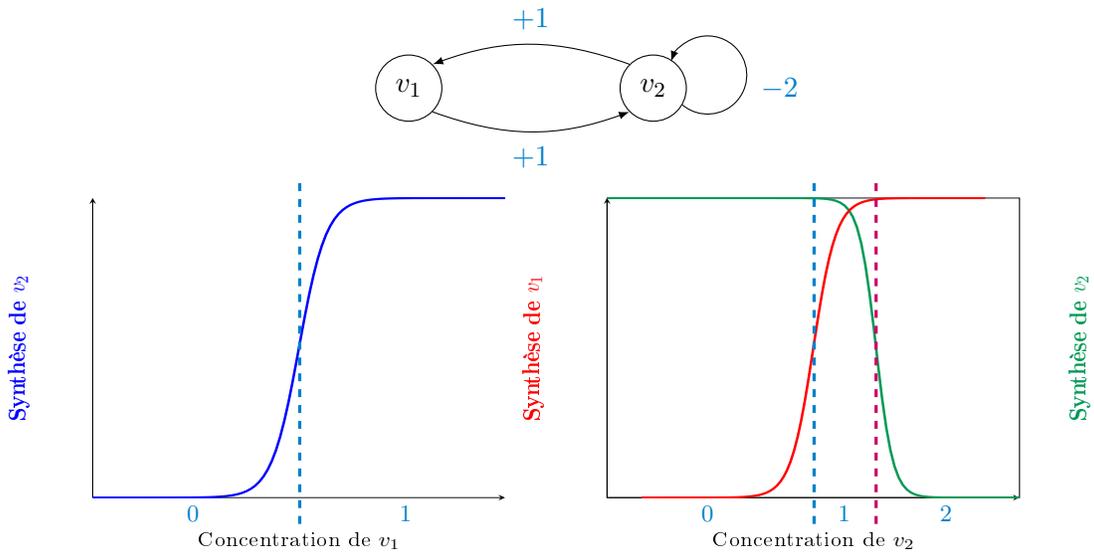


FIGURE 3.2 – Discrétisation des concentrations en niveaux qualitatifs d’expression. L’entité v_1 présente deux niveaux (0 inactif et 1 actif). L’entité v_2 présente 3 niveaux : v_2 peut ne réguler aucune de ses cibles (niveau 0), elle peut agir sur v_1 uniquement (niveau 1) ou sur v_1 et elle-même (niveau 2).

production de v_2 après influence dépend de la concentration de v_1 , et la courbe que représente le taux de production de v_2 après influence en fonction de la concentration de v_1 est une sigmoïde

plus ou moins raide. L'ensemble des concentrations de v_1 peut donc être découpé en 2 intervalles de concentrations, appelés **niveaux qualitatifs** correspondant à des espaces de concentrations homogènes où l'activité des entités ne changent pas, voir figure 3.2 page précédente. Le niveau qualitatif 0 d'une entité représente l'intervalle de concentration où l'entité est inactive tandis qu'un niveau qualitatif 1 représente l'intervalle où l'entité est active. De plus, lorsqu'une entité régule plusieurs cibles, il n'y a pas de raison pour que le seuil des sigmoïdes de ces différentes régulations soient les mêmes. Dans de tels cas, il existe plusieurs niveaux qualitatifs pour représenter l'effet de l'entité sur ses cibles. Ces entités sont dites **multivaluées**. Le niveau maximal d'une entité v est noté b_v .

Enfin, un **état qualitatif** ou état discret nommé η est la donnée d'un niveau qualitatif pour chaque entité. L'ensemble des états qualitatifs définit alors l'univers du modèle et la dynamique est représentée par un **graphe d'états**, voir figure 3.1 page précédente. Dans l'exemple de ce graphe d'influence, il existe 4 états qualitatifs (η_{v_1}, η_{v_2}) représentés par des carrés : $(0, 0)$, $(0, 1)$, $(1, 0)$ et $(1, 1)$.

Deux états qualitatifs sont appelés **états voisins** s'ils ne varient que d'une seule composante et si les niveaux d'expressions de l'entité associée ne diffèrent que de 1.

Dans cette approche discrète, il est possible d'atteindre plusieurs états voisins à partir d'un état qualitatif. C'est une caractéristique de ce cadre qui permet des comportements qu'il ne serait pas possible d'observer dans certains cadres de modélisation.

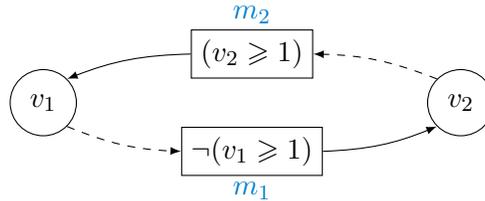


FIGURE 3.3 – Graphe d'influence avec multiplexes représentant une boucle de rétroaction négative simple.

Dans nos travaux, le graphe d'influence est complété avec les **multiplexes** qui spécifient, par une formule logique, les conditions de régulations de coopération ou de concurrence entre deux ou plusieurs entités sur une même cible [77], voir figure 3.3. Les arcs pointillés arrivant sur un multiplexe indiquent les entités présentes dans la formule logique. Ces arcs pointillés peuvent être reconstruits en regardant uniquement les formules des multiplexes. Ces arcs ne sont donc pas spécifiés dans la définition formelle des graphes d'influence. Enfin, les multiplexes agissent toujours **positivement** sur une entité cible, représentée par des arcs pleins. Ainsi, un ensemble de régulations R est définie par un ensemble V de variables, un ensemble de multiplexes M , un

ensemble d'arcs $E \subset M \times V$ et un ensemble de paramètres K . On notera $R = (V, M, E, K)$.

Le graphe d'influence de la figure 3.2 utilisé classiquement dans l'approche discrète peut être facilement transformé en un graphe avec multiplexes : pour un arc de la forme $v_1 \xrightarrow{+1} v_2$ (resp. $v_2 \xrightarrow{-1} v_2$), on ajoute un multiplexe de formule $m = v_1 \geq 1$ (resp. $m = \neg(v_2 \geq 1)$) et un arc $m \rightarrow v_2$. On peut montrer que dans ce cas, les 2 représentations sont équivalentes.

Les multiplexes agissant sur une entité sont appelées **prédécesseurs**, et l'ensemble des prédécesseurs d'une entité v est noté $R^-(v)$. Les **ressources** d'une entité dans un état donné sont les multiplexes prédécesseurs dont les formules sont satisfaites dans l'état considéré. Notons que l'inactivité d'un inhibiteur revient à activer l'élément ciblé.

Le graphe d'états permet donc de représenter la dynamique d'un système. Cette dynamique est gouvernée par des paramètres dynamiques K qui donnent le niveau qualitatif vers lequel chaque entité est attirée. Chaque **paramètre dynamique** K est indexé par une entité et un ensemble de ressources (exemples : $K_{v,\emptyset}$, $K_{v,\{m_1\}}$, etc). Ainsi, puisque ces paramètres représentent des niveaux qualitatifs, $\forall v \in V, \forall \omega \in R^-(v), K_{v,\omega} \in \llbracket 0, b_v \rrbracket$ ¹.

Des contraintes existent entre les paramètres dynamiques d'une même entité. En effet, puisque les multiplexes agissent toujours positivement sur leur entité cible, l'ajout d'une ressource sur un paramètre K ne peut pas faire diminuer sa valeur [120]. C'est la condition de Snoussi :

$$\forall v \in V, \forall \omega, \omega' \subset R^-(v), \omega \subset \omega' \Rightarrow K_{v,\omega} \leq K_{v,\omega'}$$

Ces contraintes, en plus des connaissances biologiques, peuvent permettre de déterminer les paramètres dynamiques des modèles biologiques décrits par des graphes d'influence.

C'est notamment le cas sur le graphe du cycle circadien simplifié présenté en [35]. Ce graphe comporte 3 entités : une abstraction des gènes *per* et *cry* notée G , le complexe protéique PER/-CRY noté PC et la lumière L . Le réseau de régulation défini est $R = (V, M, E, K)$ où $V = \{G, PC, L\}$, M est l'ensemble des multiplexes $\{m_G, m_{PC}, m_L\}$ dont les formules sont respectivement $(\varphi_{m_G} = G \geq 1)$, $(\varphi_{m_{PC}} = PC \geq 1)$ et $(\varphi_{m_L} = \neg(L \geq 1))$, les cibles de ces multiplexes sont définies par l'ensemble des arcs $E = \{m_G \rightarrow PC, m_{PC} \rightarrow G, m_L \rightarrow L, m_L \rightarrow PC\}$ et K est l'ensemble des paramètres dynamiques $K = \{K_{G,\{\}}, K_{G,\{m_{PC}\}}, K_{L,\{\}}, K_{L,\{m_L\}}, K_{PC,\{\}}, K_{PC,\{m_G\}}, K_{PC,\{m_L\}}, K_{PC,\{m_G, m_L\}}\}$. Enfin, les valeurs de ces 8 paramètres dynamiques sont les suivantes :

$$\begin{aligned} K_{G,\{\}} &= 0, & K_{G,\{m_{PC}\}} &= 1, \\ K_{L,\{\}} &= 0, & K_{L,\{m_L\}} &= 1, \\ K_{PC,\{\}} &= 0, & K_{PC,\{m_G\}} &= 0, & K_{PC,\{m_L\}} &= 0, & K_{PC,\{m_G, m_L\}} &= 1 \end{aligned}$$

Ces valeurs ont été obtenues par les connaissances biologiques sur le cycle circadien. Par

1. $v \in \llbracket a, b \rrbracket = \{v \in \mathbb{N} \mid a \leq v \leq b\}$.

exemple, l'horloge circadienne oscille en nuit constante, et donc la boucle entre les entités G et PC oscillent, ce qui signifie que $K_{G,\{\}} = 0$ et $K_{G,\{m_{PC}\}} = 1$.

Pour synthétiser, l'approche discrète de René Thomas présente un intérêt pour abstraire les concentrations des entités de systèmes biologiques. Cependant, le temps chronométrique joue un rôle crucial dans de nombreux systèmes biologiques et est présent dans peu de modèles discrets [118]. Nous avons donc utilisé un cadre de modélisation hybride basé sur l'approche discrète et qui permet de prendre en compte ces informations chronométriques.

3.3 Le cadre de modélisation hybride

Le cadre de modélisation hybride, comme le cadre discret, abstrait les concentrations des entités par des niveaux qualitatifs tout en permettant de manipuler le temps passé dans un état qualitatif avant d'atteindre un état voisin. Ce cadre de modélisation a été défini dans des travaux précédents au sein de l'équipe [37] dont l'intérêt réside dans la prise en compte des informations temporelles provenant d'observations qui peuvent donner lieu à des contraintes sur la vitesse d'évolution de chaque entité dans les états parcourus. Ainsi, le cadre de modélisation hybride introduit un graphe d'états similaire à celui de l'approche discrète mais rajoute des trajectoires continues des entités à l'intérieur de chaque état qualitatif.

3.3.1 Réseau de régulation hybride

Le cadre de modélisation hybride est défini pour des graphes d'influence avec multiplexes appelé aussi réseaux de régulation hybride, voir définition 3.1.

Réseau de régulation hybride

Définition 3.1: Un réseau de régulation hybride (RRH) est un quadruplet $R = (V, M, E, \mathcal{C})$ où :

- ◇ V est un ensemble fini dont les éléments sont appelés *entités* du réseau. A chaque entité $v \in V$ est associée une borne limite $b_v \in \mathbb{N}^*$.
- ◇ M est un ensemble fini dont les éléments sont appelés *multiplexes*. A chaque multiplexe $m \in M$ est associé une formule φ_m formée des atomes ($v \geq n$) où $v \in V$, $n \in \llbracket 1, b_v \rrbracket$, et des connecteurs logiques usuels \neg , \wedge , \vee et \Rightarrow .
- ◇ E est un ensemble d'arcs de la forme $(m \rightarrow v) \in M \times V$.
- ◇ $\mathcal{C} = \{C_{v,\omega,n}\}$ est une famille de nombres réels indexés par un triplet (v, ω, n) où $v \in V$, ω est un sous-ensemble de $R^-(v)$ où $R^-(v) = \{m \mid (m \rightarrow v) \in E\}$, c'est-à-dire, ω est un ensemble de prédécesseurs de v , et $n \in \llbracket 0, b_v \rrbracket$. $C_{v,\omega,n}$ est appelé la *célérité* de v pour ω au niveau n et décrit l'évolution de l'entité v .

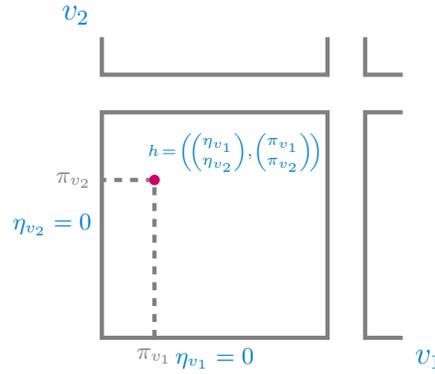


FIGURE 3.4 – Représentation d'un état hybride dans un état qualitatif.

3.3.2 Etat d'un réseau de régulation hybride

Dans l'approche hybride, chaque état qualitatif peut être vu comme un hypercube $[0, 1]^n$ où n est le nombre d'entités dans le graphe d'influence. Un état du système est caractérisé par un état qualitatif et par une position précise à l'intérieur de cet état. Ces états sont appelés **états hybrides** et sont représentés par des couples $h = (\eta, \pi)$ où η représente les niveaux qualitatifs des entités et π représente les positions exactes à l'intérieur de l'état qualitatif. η est appelé **état qualitatif** alors que π est appelée **partie fractionnaire**, voir définition 3.2. L'exemple de la figure 3.4 illustre un état qualitatif $(\eta_{v_1}, \eta_{v_2}) = (0, 0)$ dans lequel se trouve un état hybride h tel que $h = \left(\begin{pmatrix} \eta_{v_1} \\ \eta_{v_2} \end{pmatrix}, \begin{pmatrix} \pi_{v_1} \\ \pi_{v_2} \end{pmatrix} \right)$.

Etat d'un RRH

Définition 3.2: Soit $R = (V, M, E, C)$ un RRH. Un *état hybride* de R est un couple $h = (\eta, \pi)$ où

- ◊ η est une fonction de V dans \mathbb{N} telle que pour tout $v \in V$, $0 \leq \eta(v) \leq b_v$;
- ◊ π est une fonction de V dans l'intervalle des nombres réels $[0, 1]$.

η est appelé *état qualitatif* ou *état discret* de h tandis que π est appelé *partie fractionnaire*.

Pour simplifier, nous notons pour la suite $\eta_v = \eta(v)$ et $\pi_v = \pi(v)$. Nous dénotons S l'ensemble des états hybrides de R . Quand il n'y a pas d'ambiguïté, nous utilisons souvent η et π sans mention explicite de h .

Dans chaque état qualitatif, la trajectoire du modèle est décrite par les célérités de chacune des entités à l'intérieur de cet état. Chaque célérité dépend de l'ensemble des ressources d'une entité qui la pilote. Les **célérités** représentent les vitesses d'évolution des entités en fonction du niveau qualitatif de l'entité et de l'ensemble de ses ressources. Les ressources des célérités

sont définies de la manière suivante :

Ressources

Définition 3.3: Soit $R = (V, M, E, \mathcal{C})$ un RRH et soit $v \in V$. La relation de satisfaction $h \models \varphi$, où $h = (\eta, \pi)$ est un état hybride de R et φ est une formule de multiplexe, est définie inductivement comme suit :

- ◇ Si φ est l'atome $(v \geq n)$ avec $n \in \llbracket 1, b_v \rrbracket$, alors $h \models \varphi$ si et seulement si $\eta_v \geq n$;
- ◇ La signification usuelle des connecteurs logiques est utilisée.

L'ensemble des *ressources* d'une entité v à un état h est défini par : $\rho(h, v) = \{m \in R^-(v) \mid h \models \varphi_m\}$, c'est-à-dire, les multiplexes prédécesseurs de v dont la formule est satisfaite.

Par abus de langage, nous utilisons aussi la notation $\rho(\eta, v)$ plutôt que $\rho(h, v)$ puisque les ressources à l'intérieur d'un état qualitatif ne change pas.

De plus, les célérités ne dépendent pas que des ressources mais aussi des niveaux qualitatifs. En effet, quand une entité est soumise à un ensemble de ressources, son niveau d'expression tend vers un état d'équilibre. Selon le niveau qualitatif, on va vers l'état d'équilibre plus ou moins rapidement ce qui explique que la célérité dépend non seulement de ω mais aussi de n .

Les célérités d'une même entité sont contraintes de la manière suivante :

$$\forall v \in V, \forall \omega \subset R^-(v), \forall n \in \llbracket 0, b_v \rrbracket, \quad C_{v,\omega,n} = 0 \Rightarrow \begin{cases} \forall i \in \llbracket n+1, b_v \rrbracket & C_{v,\omega,i} < 0 \\ \forall i \in \llbracket 0, n-1 \rrbracket & C_{v,\omega,i} > 0 \end{cases}$$

$$\forall v \in V, \forall \omega \subset R^-(v), \forall k \in \llbracket 0, b_v - 1 \rrbracket, \quad C_{v,\omega,k} \times C_{v,\omega,k+1} \geq 0.$$

Ces contraintes indiquent que les célérités d'une entité v qui diffèrent uniquement par leurs niveaux d'expressions se dirigent vers un même état d'équilibre (les entités sont soumises aux mêmes ressources), ou bien les célérités sont de même signe car elles se dirigent vers un même état d'équilibre situé à l'extérieur des états qualitatifs, voir figure 3.5 page ci-contre. En effet, puisque les ressources ne changent pas, l'effet des prédécesseurs sur l'entité cible reste inchangé.

3.3.3 Délai de contact

A partir d'un état hybride donné, nous souhaitons identifier le temps nécessaire à chaque entité pour atteindre un bord. Ce temps est nommé **délai de contact** de l'entité v . Il dépend de la distance qu'il reste à parcourir et de la célérité de cette entité, voir définition 3.4 page suivante. Cette information est essentielle pour extraire des contraintes sur les célérités.

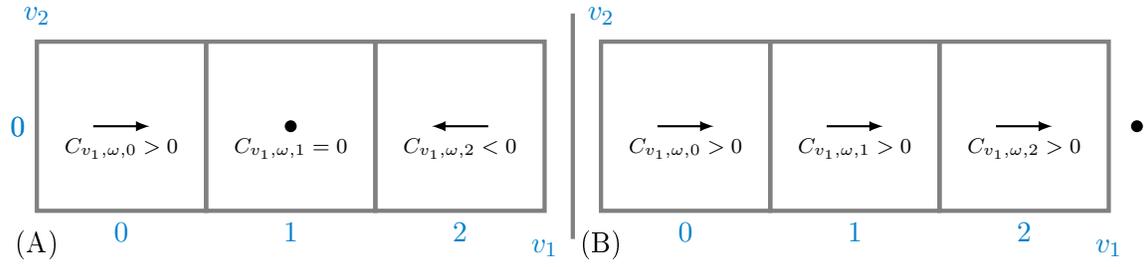


FIGURE 3.5 – Etat d'équilibre pour des célérités d'une entité. Les célérités définies diffèrent uniquement par rapport à leurs niveaux d'expressions. (A) L'état d'équilibre est localisé à l'intérieur d'un état qualitatif. (B) L'état d'équilibre est situé à l'extérieur des états qualitatifs.

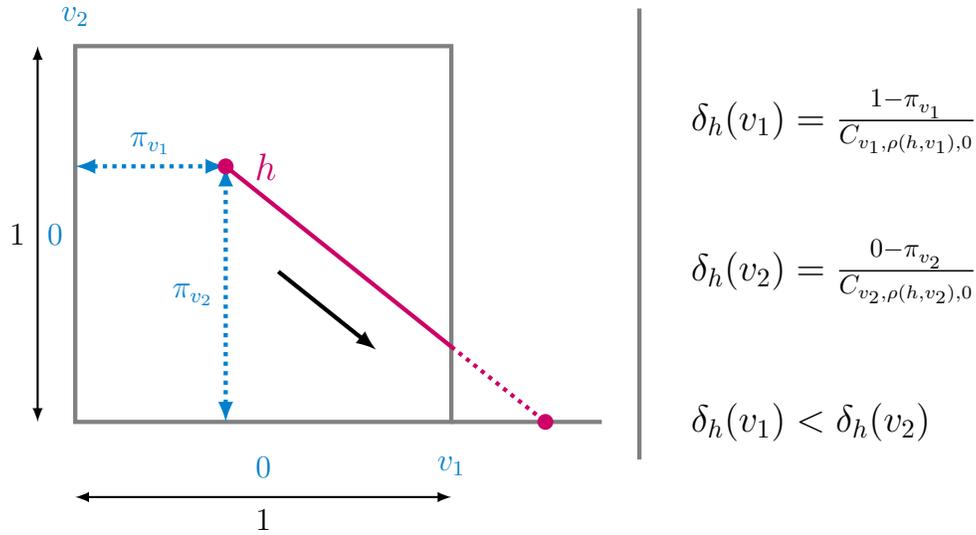


FIGURE 3.6 – Délai de contact dans un état qualitatif.

Délai de contact

Définition 3.4: Soit $R = (V, M, E, C)$ un réseau de régulation hybride, v une entité de V et $h = (\eta, \pi)$ un état hybride. Nous notons $\delta_h(v)$, appelé le *délai de contact* de v en h , le temps nécessaire à v pour atteindre le bord de l'état qualitatif. δ_h est la fonction de V dans $\mathbb{R}^+ \cup \{+\infty\}$ définie par :

- ◇ Si $C_{v, \rho(h, v), \eta_v} = 0$ alors $\delta_h(v) = +\infty$;
- ◇ Si $C_{v, \rho(h, v), \eta_v} > 0$ (resp. < 0) alors $\delta_h(v) = \frac{1 - \pi_v}{C_{v, \rho(h, v), \eta_v}}$ (resp. $\frac{0 - \pi_v}{C_{v, \rho(h, v), \eta_v}}$).

La figure 3.6 illustre le calcul du délai de contact des entités v_1 et v_2 . A partir d'un état hybride h , le délai de contact avant d'atteindre le bord supérieur de v_1 est $\delta_h(v_1) = \frac{1 - \pi_{v_1}}{C_{v_1, \rho(h, v_1), 0}}$,

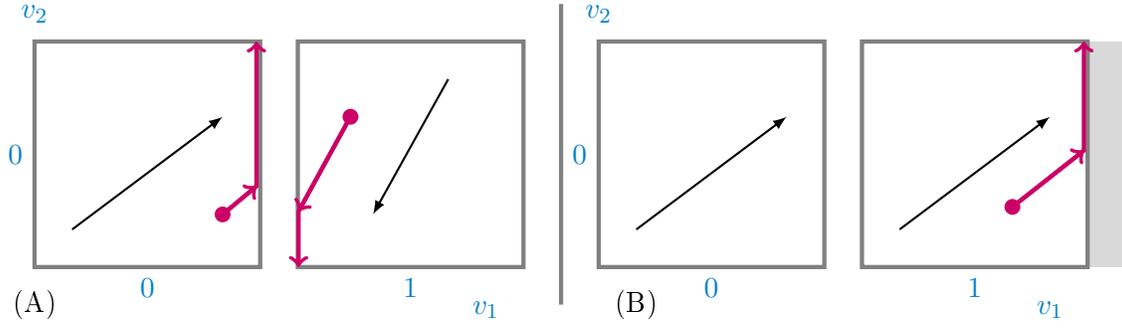


FIGURE 3.7 – Murs interne et externe de l’entité v_1 . (A) Mur interne. (B) Mur externe

tandis que le délai de contact avant d’atteindre le bord inférieur de v_2 est $\delta_h(v_2) = \frac{0 - \pi_{v_2}}{C_{v_2, \rho(h, v_2), 0}}$. Dans cette figure, le bord de v_1 est atteint avant le bord de v_2 , indiquant que $\delta_h(v_1) < \delta_h(v_2)$. Cette information permet d’apporter des contraintes entre les célérités des différentes entités dans un même état qualitatif.

3.3.4 Murs interne et externe

Atteindre un bord d’une entité ne suffit pas à le franchir. En effet, une entité peut franchir un bord si elle ne rencontre pas d’impossibilité de le franchir. Lorsqu’il y a impossibilité, on dit que la variable fait face à un **mur interne** ou à un **mur externe** comme indiqué en définition 3.5.

Murs interne et externe

Définition 3.5: Soit $R = (V, M, E, C)$ un réseau de régulation hybride RRH, $v \in V$ une entité et $h = (\eta, \pi)$ un état hybride.

1. v fait face à un mur externe en h si :

$$((C_{v, \rho(h, v), \eta_v} < 0) \wedge (\eta_v = 0)) \vee ((C_{v, \rho(h, v), \eta_v} > 0) \wedge (\eta_v = b_v))$$

2. Soit $h' = (\eta', \pi')$ un autre état hybride tel que $\eta'_v = \eta_v + \text{sgn}(C_{v, \rho(h, v), \eta_v})$ et $\eta'_u = \eta_u$ pour tout $u \in V, u \neq v$. v fait face à un mur interne en h si :

$$\text{sgn}(C_{v, \rho(h, v), \eta_v}) \times \text{sgn}(C_{v, \rho(h', v), \eta'_v}) = -1,$$

avec sgn la fonction signe usuelle, $\rho(h, v)$ les ressources de v dans h .

On note $\text{se}(h)$ (pour *sliding entities*.) l’ensemble des **entités glissantes**, c’est-à-dire qui font face à un mur interne ou externe en h .

Dans la figure 3.7, l’entité v_1 fait face à un mur interne en (A) et à un mur externe en (B) car elle ne peut pas franchir son bord. Dans de telles situations, la trajectoire glisse selon la

composante v_2 en fonction de sa célérité.

3.3.5 Entité changeante *first*

A partir d'un état hybride h , les entités atteignant leur bord en premier présentent le délai de contact minimum dans l'état. Cependant, les entités faisant face à un mur ne peuvent pas franchir de seuil. C'est pourquoi on définit l'ensemble des entités qui peuvent franchir leur seuil en premier par l'ensemble **first** :

Entités changeantes *first*

Définition 3.6: Soit $R = (V, M, E, C)$ un RRH et soit $h = (\eta, \pi)$ un état hybride. L'ensemble des *entités changeantes first* est défini par :

$$\text{first}(h) = \{v \in V \setminus \text{se}(h) \mid \delta_h(v) \neq +\infty \wedge \forall u \in V \setminus \text{se}(h), \delta_h(u) \geq \delta_h(v)\} .$$

De plus, δ_h^{first} dénote le temps passé dans l'état qualitatif de h lorsqu'on démarre de h : pour tout $v \in \text{first}(h)$, $\delta_h^{\text{first}} = \delta_h(v)$, ou bien $\delta_h^{\text{first}} = +\infty$ si $\text{first}(h) = \emptyset$.

D'après la définition, puisque les entités dans *first* ne peuvent pas être des entités glissantes, on en déduit, pour un état hybride h , que $\text{first}(h) \cap \text{se}(h) = \emptyset$. Les entités dans *first* représentent alors les entités qui atteignent leur bord en premier et qui peuvent le franchir.

3.3.6 Graphe d'états hybrides : transitions continues et discrètes

La notion d'état hybride permet de définir deux types de transition : les **transitions continues** et les **transitions discrètes**. La transition continue $h \rightarrow h'$ avec $h = (\eta, \pi)$ et $h' = (\eta, \pi')$ indique qu'il existe une trace partant de l'état qualitatif courant qui mène à h' en partant de h . La transition discrète $h \rightarrow h'$ avec $h = (\eta, \pi)$ et $h' = (\eta', \pi')$ indique qu'il existe une trace dans l'état qualitatif courant menant à un état qualitatif voisin.

Nous pouvons à présent définir l'espace des états hybrides représentant la dynamique.

Graphe d'états hybrides

Définition 3.7: Soit $R = (V, M, E, C)$ un RRH. Nous notons $\mathcal{R} = (S, cT, dT)$ le *graphe d'états hybrides* de R où S est l'ensemble des états hybrides, et cT (resp. dT) est l'ensemble des *transitions continues* (resp. *discrètes*) :

1. Il existe une unique transition continue dans cT de l'état $h = (\eta, \pi)$ vers l'état $h' = (\eta', \pi')$ si et seulement si :
 - (a) Soit $\text{first}(h) \neq \emptyset$ et il existe une entité $v \in \text{first}(h)$ tel que :
 - i. $\delta_h(v) \neq 0$, où $\delta_h(v)$ est appelée la *durée* de la transition (continue),
 - ii. $\eta' = \eta$ et $\pi'_u = \begin{cases} 0 & \text{si } C_{u,\rho(h,u),\eta_u} < 0 \\ 1 & \text{si } C_{u,\rho(h,u),\eta_u} > 0 \end{cases}$ pour tout $u \in (\text{first}(h) \cup \sigma)$
où $\sigma = \{x \in \text{se}(h) \mid \delta_h(x) \leq \delta_h^{\text{first}}\}$,
 - iii. $\forall z \in V \setminus (\text{first}(h) \cup \sigma)$, alors $\pi'_z = \pi_z + \delta_h(v) \times C_{z,\rho(h,z),\eta_z}$.
 - (b) Soit $\text{first}(h) = \emptyset$ (signifiant que chaque entité v soit atteint un état d'équilibre $C_{v,\rho(h,v),\eta_v} = 0$, soit fait face à un mur $v \in \text{se}(h)$) et :
 - i. $\forall v \in \text{se}(h)$, $\pi'_v = \begin{cases} 0 & \text{si } C_{v,\rho(h,v),\eta_v} < 0 \\ 1 & \text{si } C_{v,\rho(h,v),\eta_v} > 0 \end{cases}$
 - ii. $\forall u \notin \text{se}(h)$, $\pi'_u = \pi_u$ (puisque dans ce cas $C_{u,\rho(h,u),\eta_u} = 0$).
2. Il existe une transition discrète dans dT de l'état $h' = (\eta', \pi')$ vers l'état $h'' = (\eta'', \pi'')$ si et seulement si il existe une entité $v \in \text{first}(h')$ telle que :
 - (a) $\delta_{h'}(v) = 0$, où $\delta_{h'}(v)$ est appelée la *durée* de la transition (discrète),
 - (b) $\eta''_v = \eta'_v + \text{sgn}(C_{v,\rho(h',v),\eta'_v})$ et $\pi''_v = \begin{cases} 0 & \text{si } C_{v,\rho(h',v),\eta'_v} > 0 \\ 1 & \text{si } C_{v,\rho(h',v),\eta'_v} < 0 \end{cases}$,
 - (c) $\forall u \in V \setminus \{v\}$, $\eta''_u = \eta'_u$ et $\pi''_u = \pi'_u$.

Les états à partir desquels il n'existe pas de transitions (discrètes ou continues) sont appelés des **états stables**.

Une transition continue portant d'un état hybride h définit un nouvel état hybride : l'état qualitatif du nouvel état hybride est le même, mais les parties fractionnaires de chaque entité ont été modifiées. Une transition continue $h \rightarrow h'$ est possible s'il existe une entité changeante de l'ensemble *first* menant à atteindre le bord de cette entité, et lorsqu'il n'en existe pas, l'état d'arrivée de la transition continue est tel que les entités de $\text{se}(h)$ atteignent leur bord et les entités de célérités nulles ne bougent pas.

Enfin, une transition discrète est possible si deux conditions sont respectées : une entité doit

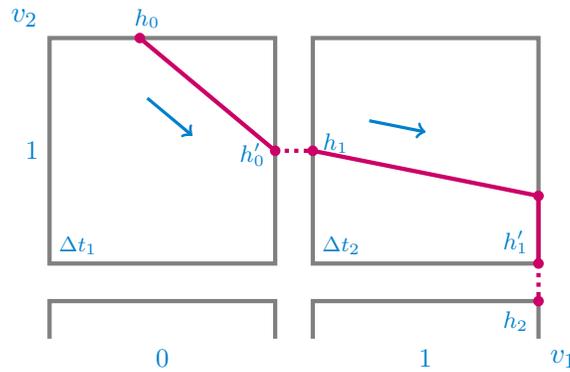


FIGURE 3.8 – Trace biologique avec des transitions continues et discrètes. Les transitions continues sont représentées en trait rose plein et les transitions discrètes en trait rose pointillé. Δt_1 et Δt_2 représentent les temps passés dans les états qualitatifs et les vecteurs représentent les vecteurs célérités.

être sur un de ses bords dans l'état qualitatif courant (le délai de contact de cette entité est nulle) et cette entité doit pouvoir franchir ce bord afin d'atteindre un état voisin (cette entité ne fait pas face à un mur interne ou externe).

Dans l'exemple de la figure 3.8, selon les vecteurs célérités dessinés, il existe une transition continue $h_0 \rightarrow h'_0$ dans l'état qualitatif $(\eta_{v_1}, \eta_{v_2}) = (0, 1)$ suivi d'une transition discrète $h'_0 \rightarrow h_1$ qui permet d'atteindre l'état qualitatif $(1, 1)$. Une nouvelle transition continue $h_1 \rightarrow h'_1$ est possible : elle comporte un glissement de la composante v_2 car l'entité v_1 atteint un mur externe sur son bord supérieur. Enfin, une transition discrète $h'_1 \rightarrow h_2$ est possible et diminue le niveau qualitatif de v_2 .

Ainsi, dans le cadre de modélisation hybride, la trajectoire est représentée comme une succession de transitions alternant transitions continues et transitions discrètes. C'est une trajectoire linéaire par morceau qui peut mener à un comportement non-déterministe. En effet, si les vecteurs célérités permettent à la trajectoire d'atteindre plusieurs bords d'entités changeantes de *first* en même temps, plusieurs transitions discrètes sont possibles, voir figure 3.9 page suivante (B).

3.4 Exemple sur la boucle négative

Reprenons l'exemple du graphe d'influence de la figure 3.3 page 46 définissant une boucle de rétrocontrôle négative entre les entités v_1 et v_2 . C'est un phénomène couramment observé en biologie où une entité active son inhibiteur. Dans cet exemple, chaque entité présente deux niveaux d'expressions : le niveau qualitatif 0 où l'entité est inactive et le niveau 1 où l'entité est

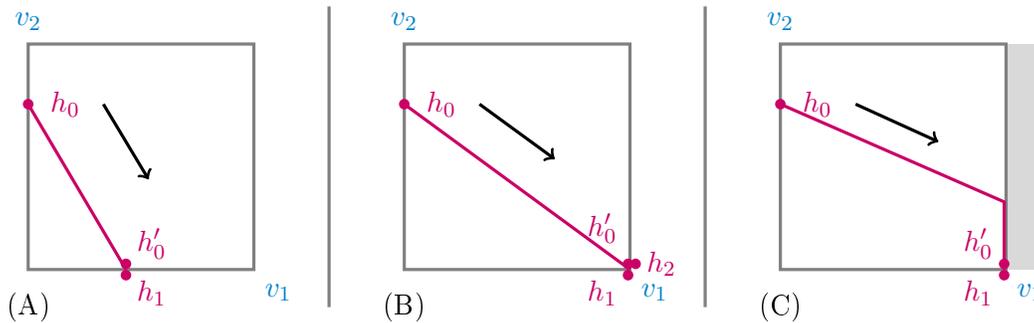


FIGURE 3.9 – Transitions continues et discrètes. Dans les différents cas, il existe une transition continue $h_0 \rightarrow h'_0$. (A) Les vecteurs célérités autorisent la transition discrète $h'_0 \rightarrow h_1$ sans glissement. (B) Depuis l'état hybride h'_0 , deux transitions discrètes sont possibles : $h'_0 \rightarrow h_1$ et $h'_0 \rightarrow h_2$. (C) La zone grisée présente un mur interne ou externe. La seule transition discrète possible est $h'_0 \rightarrow h_1$.

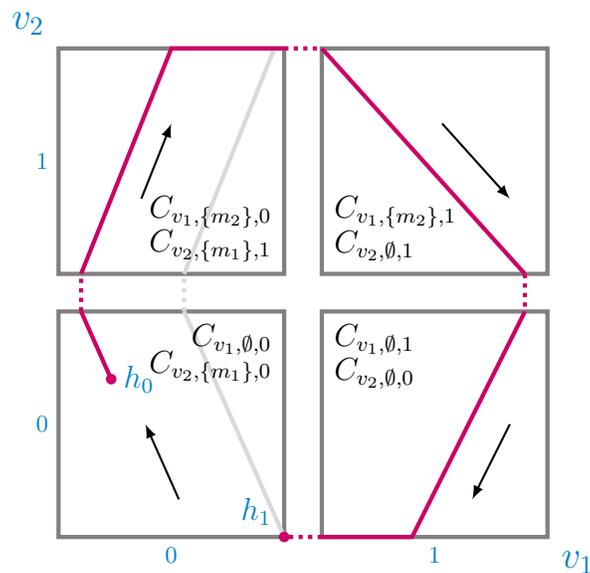


FIGURE 3.10 – Une dynamique possible associée au graphe d'influence de la figure 3.3 page 46. A partir d'un état hybride initial h_0 , il est possible d'observer la dynamique des entités (trait rose) dans les différents états qualitatifs du système. Le trait grisé montre le cycle limite.

active. L'ensemble des prédécesseurs possibles de v_1 est réduit au singleton ne contenant que le multiplexe noté m_2 et celui de v_2 est réduit à $\{m_1\}$.

Le cadre de modélisation hybride permet de définir le graphe d'états contenant 4 états qualitatifs : $(0, 0)$, $(0, 1)$, $(1, 0)$ et $(1, 1)$, voir figure 3.10 page ci-contre. De plus, les célérités gouvernant ces états qualitatifs sont au nombre de 8 :

$$\begin{array}{cccc} C_{v_1, \emptyset, 0}, & C_{v_1, \emptyset, 1}, & C_{v_1, \{m_2\}, 0}, & C_{v_1, \{m_2\}, 1} \\ C_{v_2, \emptyset, 0}, & C_{v_2, \emptyset, 1}, & C_{v_2, \{m_1\}, 0}, & C_{v_2, \{m_1\}, 1} \end{array}$$

Les signes de toutes les célérités peuvent être instanciés de façon intuitive. Quand v_2 est au niveau 1 (actif), le multiplexe m_2 agit car v_2 joue son rôle d'activateur sur v_1 : l'entité v_1 augmente ($C_{v_1, \{m_2\}, 0} > 0$ et $C_{v_1, \{m_2\}, 1} > 0$). Si v_2 est au niveau 0, m_2 est inactif et donc l'entité v_1 diminue ($C_{v_1, \emptyset, 0} < 0$ et $C_{v_1, \emptyset, 1} < 0$). Quand v_1 est au niveau 0 (inactif), le multiplexe m_1 agit car v_1 ne peut pas jouer son rôle d'inhibiteur sur l'entité v_2 : l'entité v_2 augmente ($C_{v_2, \{m_1\}, 0} > 0$ et $C_{v_2, \{m_1\}, 1} > 0$). A l'inverse, lorsque v_1 est au niveau 1 (m_1 est inactif), l'entité v_2 diminue ($C_{v_2, \emptyset, 0} < 0$ et $C_{v_2, \emptyset, 1} < 0$).

En considérant des valeurs des célérités choisies au hasard mais qui satisfont les inégalités précédentes, on peut étudier les trajectoires du modèle, voir figure 3.10 page précédente dans laquelle les vecteurs dessinés décrivent les célérités des entités dans les différents états qualitatifs. A partir d'un état hybride h_0 situé dans l'état qualitatif $(\eta_{v_1}, \eta_{v_2}) = (0, 0)$, la transition continue partant de h_0 mène à atteindre le bord supérieur de v_1 (trait rose plein). La transition discrète (trait rose pointillé) qui en résulte permet d'atteindre l'état qualitatif $(1, 0)$. La transition continue dans l'état qualitatif $(0, 1)$ présente un glissement de v_2 (v_2 atteint un mur externe) puis la transition discrète permet d'atteindre l'état qualitatif $(1, 1)$. L'alternance de transitions continues et discrètes se poursuit en passant encore par un glissement de v_2 en $(1, 0)$ (v_2 fait face à un mur externe représentant une dégradation totale) jusqu'à atteindre h_1 . La trajectoire peut se poursuivre (trait gris plein) jusqu'à atteindre le bord supérieur de v_2 dans l'état qualitatif $(0, 1)$. Un **cycle limite** est atteint, c'est-à-dire que la trajectoire va repasser par les mêmes transitions continues et discrètes, ce qui décrit un comportement oscillatoire infini.

3.5 Positionnement par rapport à nos objectifs

Au départ, notre objectif était de modéliser les systèmes biologiques à comportement oscillatoire par un modèle informatique. Pour cela, nous avons utilisé un cadre de modélisation hybride [37] pour représenter ces systèmes biologiques. Les graphes d'états décrivent la dynamique du système lorsque les paramètres dynamiques sont connus. Ces paramètres dynamiques nommés célérités ne sont malheureusement pas connus et que très rarement mesurables expérimentalement. Ils doivent donc être identifiés pour que le modèle soit utilisable.

A la différence des paramètres dynamiques K du modèle discret, les célérités dépendent aussi des niveaux d'expressions des entités. Cela signifie que le nombre de paramètres dynamiques à identifier est plus important dans le cadre de modélisation hybride que dans l'approche discrète pour un même graphe d'influence. Par exemple, le modèle discret de la boucle négative de la section précédente comporte 4 paramètres K alors que le modèle hybride comporte 8 célérités. Si nous avons augmenté le nombre de paramètres dynamiques à identifier, nous avons aussi augmenté le nombre de contraintes sur ces paramètres : les informations temporelles, les saturations ou dégradations totales de protéines sont des éléments essentiels contraignant l'évolution du système et les paramètres dynamiques.

Pour identifier les célérités, nous avons développé une méthode formelle basée sur la logique de Hoare afin de pouvoir contraindre les célérités à partir des traces biologiques expérimentales. Cette approche a été développée car la logique de Hoare dans l'approche discrète de René Thomas a montré toute sa puissance [17] et a été utilisée dans d'autres systèmes hybrides [69]. C'est l'enjeu du prochain chapitre.

Chapitre 4

Construction des contraintes sur les paramètres

En comportement physiologique normal, la dynamique d'un système biologique est généralement stable, et présente souvent des oscillations. Le modèle d'un tel système doit donc pouvoir représenter ces comportements sans avoir à modifier au cours du temps les paramètres qui contrôlent sa dynamique. Néanmoins, ces paramètres sont contraints selon l'évolution de la trajectoire. Ce chapitre décrit la construction des contraintes sur les paramètres dynamiques de systèmes biologiques pour que l'évolution des modèles informatique corresponde aux observations expérimentales.

Après avoir rappelé les notions d'automates hybrides, de vérification d'une propriété comme par exemple l'atteignabilité d'un état, et après avoir expliqué la logique de Hoare pour la vérification de modèles discrets, nous introduisons une *logique de Hoare* spécifiquement conçue pour construire les contraintes sur les automates hybrides que nous utilisons. L'idée est de contraindre les paramètres dynamiques de ces automates hybrides permettant de rendre compte des observations expérimentales d'un système biologique étudié. Enfin, nous montrerons la correction ainsi que la complétude sémantique de la logique de Hoare *hybride*, puis nous détaillerons un exemple d'utilisation de cette logique de Hoare sur le graphe d'influence représentant une boucle de rétroaction négative.

4.1 Vérification des automates hybrides

4.1.1 Rappel sur les automates hybrides et les méthodes formelles

Un **automate hybride** est un modèle de systèmes dynamiques dans lesquels coexistent des variables discrètes et continues. Il est souvent représenté par un graphe dont les nœuds

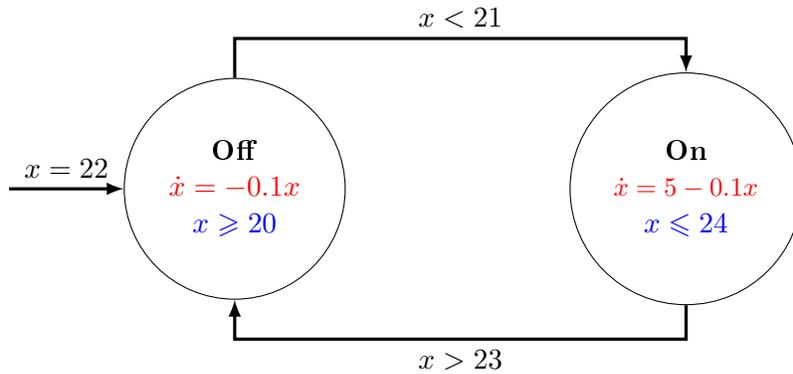


FIGURE 4.1 – Exemple d'automate hybride modélisant un thermostat. A l'intérieur d'un état qualitatif, les flux sont représentés en rouge et les invariants en bleu. Les arcs sont étiquetés par les conditions rendant le saut possible. La flèche entrante sur le nœud *Off* représente l'entrée dans l'automate.

représentent un ensemble fini d'états qualitatifs et les arcs les transitions possibles entre ces états [63]. A noter que dans les automates hybrides, les états qualitatifs sont souvent appelés *locations*. Les automates hybrides permettent de représenter :

- ◊ des comportements discrets modélisés par un automate à états finis,
- ◊ des comportements continus dans le temps, couramment modélisés par des systèmes d'équations différentielles.

La dynamique des automates hybrides est obtenue en alternant des **transitions continues** et des **transitions discrètes**. Dans un état qualitatif du système, la transition continue assure l'évolution dans le temps des variables continues de l'automate selon les formules en jeu (généralement des équations différentielles) : c'est la notion de **flux** (ou *flow* en anglais). La transition discrète, quand à elle, assure le passage d'un état qualitatif du système à un autre, plus communément appelée **saut** (ou *jump*), voir figure 4.1.

De plus, les états qualitatifs du système peuvent être munis d'**invariants** qui donnent des conditions pour pouvoir rester dans l'état qualitatif courant.

Dans l'exemple du thermostat de la figure 4.1, l'état initial du système correspond au sommet nommé *Off* où la température x vaut 22 degrés. Le flux dans ce nœud indique que la température diminue à un taux de $-0.1x$ degrés par unités de temps et le système peut rester dans cet état qualitatif tant que la température est supérieure à 20 (invariant $x \geq 20$). Lorsque la température passe en dessous de 21 degrés, la condition de saut $x < 21$ aussi appelée **garde** est respectée et la transition permet au système d'atteindre l'état étiqueté *On* (si l'invariant est respecté également). Ainsi, tant que l'invariant de l'état qualitatif courant est respecté, les variables du système sont contrôlées par le flux et lorsque la condition d'un saut est satisfaite, le système

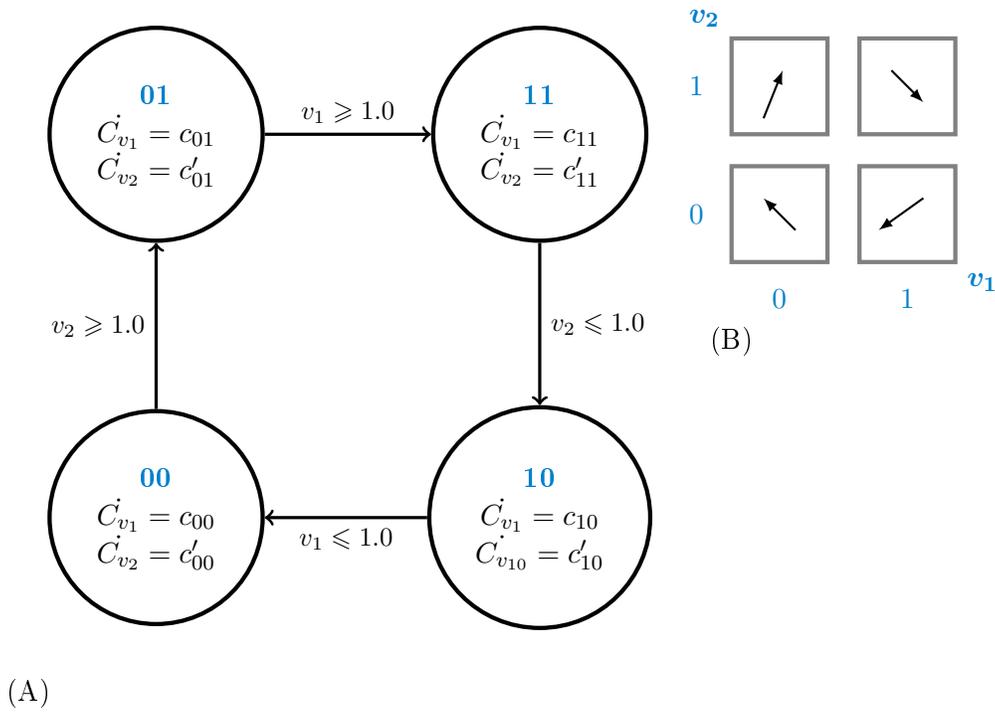


FIGURE 4.2 – Vue partielle d’automate hybride. (A) représente l’automate associé au graphe d’états détaillé en (B). Dans chaque état du système, les flux sont représentés par les célérités \dot{C}_v signifiant $C_{v,\omega,n}$ avec ω les ressources de v dans l’état et n le niveau d’expression de v . Ces flux sont tous constants (c_{\dots} et c'_{\dots}). Les invariants (non représentés) définissent les intervalles des parties fractionnaires définissant l’état qualitatif : $\forall v \in V, \pi_v \in [0, 1]$. Les sauts sont tels que $v \geq \eta_v + \pi_v$ (resp. $v \leq \eta_v - \pi_v$) pour une incrémentation (resp. décrémentation) de v . Notons que, pour des raisons pédagogiques évidentes, nous n’avons pas représenté les glissements.

peut sauter vers un nouvel état qualitatif. Le processus est alors réitéré dans le nouvel état.

Dans les automates hybrides que nous utilisons, chaque état du système fait évoluer les parties fractionnaires des différentes entités, voir figure 4.2. Les flux sont des constantes dans chaque état du système. Les invariants définissent les intervalles des parties fractionnaires qui représentent l’état qualitatif : $\forall v \in V, \pi_v \in [0, 1]$. Pour réaliser une transition discrète, les conditions de sauts sont des contraintes de la forme $v \geq \eta_v + \pi_v$ (resp. $v \leq \eta_v - \pi_v$) pour une incrémentation (resp. décrémentation) de l’entité v . Une fois que des valeurs sont données aux célérités, un automate hybride est construit et il est alors possible d’utiliser les outils classiques des automates.

Les **méthodes formelles** utilisent des logiques mathématiques basées sur la sémantique de programme informatique afin de montrer que le programme est conforme à sa spécification. Les méthodes formelles se regroupent en différentes catégories :

- ◊ l’*analyse statique* qui sur-approxime les états accessibles du système par interprétation

abstraite,

- ◇ le *model checking* qui vérifie des propriétés de logiques temporelles par une exploration exhaustive de l'espace d'états,
- ◇ la *vérification déductive* qui vérifie les systèmes par preuve.

Il est aussi important de valider des automates hybrides car ils peuvent représenter le comportement de systèmes présents dans la vie de tous les jours. Par exemple, le système de freinage dans les transports ferroviaires [107] ou encore la gestion de transactions en temps-réel. Plus généralement, tous les logiciels de contrôle dans le temps peuvent être modélisés par des automates hybrides.

Parmi les propriétés de ces systèmes, **l'atteignabilité** est intéressante à vérifier. Il s'agit de savoir si un état ou un ensemble d'états est atteignable ; cette propriété est largement utilisée pour prouver la sûreté des systèmes. La difficulté de cette preuve varie selon les classes d'automates hybrides.

4.1.2 Classes d'automates hybrides et décidabilité/indécidabilité

La difficulté du problème de l'atteignabilité est dépendante de la classe d'automates hybrides. Nous faisons ici un résumé de quelques résultats liés à la question de l'atteignabilité afin de pouvoir situer les automates hybrides introduits au chapitre précédent (nommés **AH** par la suite) par rapport aux automates hybrides existants.

Les automates hybrides linéaires

Les **automates hybrides linéaires** sont des automates où les flux, les invariants et les gardes sont linéaires [4]. Plus précisément, les flux sont de la forme $\dot{x} = c$ avec c une constante (menant à des trajectoires rectilignes), et les invariants et les gardes sont représentés par des comparaisons d'une somme pondérée de plusieurs variables avec une constante. Néanmoins, ces fortes hypothèses ne permettent pas de conclure sur la difficulté de la question d'atteignabilité, puisque cette classe englobe différentes sous-classes pour lesquelles le problème est soit indécidable, soit décidable.

Les automates hybrides rectangulaires

Dans les **automates hybrides rectangulaires**, les invariants associés à chacun des états qualitatifs sont de la forme $\bigwedge x_i \in [a_i, b_i]$ où les x_i sont les variables réelles, les a_i et b_i des constantes de \mathbb{R} [65]. Les invariants et les gardes de cette classe d'automates peuvent être vus comme des produits cartésiens d'intervalles appelés **région rectangulaire**. Les automates rectangulaires sont dits **initialisés** si, lors d'une transition discrète entre deux états qualitatifs i

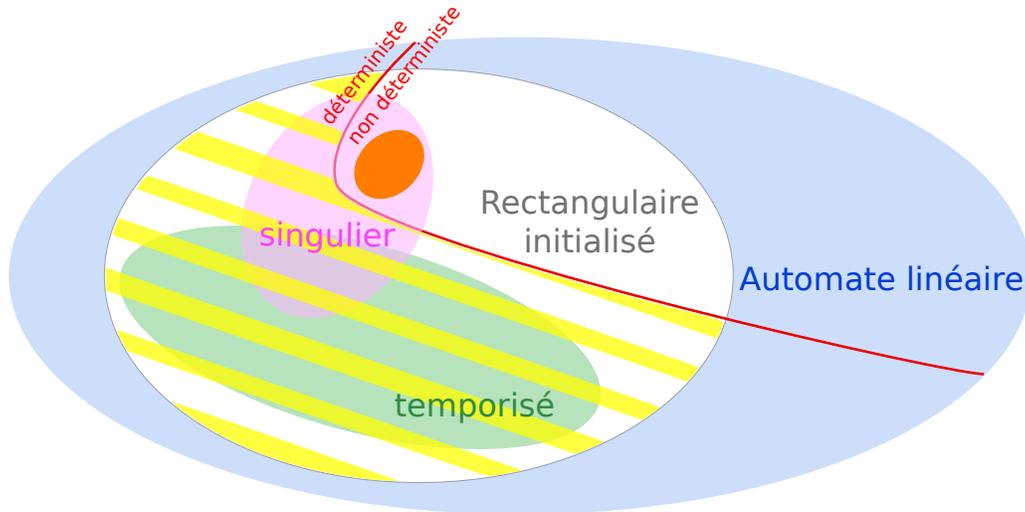


FIGURE 4.3 – Automates hybrides linéaires et atteignabilité. Les sous-ensembles représentent des classes d’automates hybrides pour lesquelles on sait si le problème d’atteignabilité est décidable ou non. Dans la classe des automates hybrides représentés en orange, à la base de nos travaux, l’indécidabilité du problème de l’atteignabilité reste une conjecture.

et j , les variables dont $\text{flux}_i \neq \text{flux}_j$ sont réinitialisées, c’est-à-dire qu’on affecte une valeur à chacune de ces variables durant la transition discrète. Contrairement aux automates rectangulaires non initialisés, l’atteignabilité pour ce type d’automate a été prouvé PSPACE complet si, dans un état qualitatif, les invariants ne comparent pas les variables entre elles. Les automates hybrides rectangulaires sont donc une sous-classe des automates hybrides linéaires où les régions linéaires sont remplacées par des régions rectangulaires.

Les automates rectangulaires contiennent différentes catégories d’automates, voir figure 4.3 :

- ◊ les **automates singuliers** dans lesquels les sauts sont déterministes (pour un état initial, le saut suivant est unique) et tous les flux des variables sont des réels finis,
- ◊ les **automates temporisés** dans lesquels les sauts sont déterministes et tous les flux des variables sont des réels tous égaux : les variables évoluent toutes à la même vitesse, et peuvent être vues comme une horloge (*clock* en anglais) à valeur réelle.

Dans ses travaux, Henzinger a prouvé que le problème de l’atteignabilité dans les automates singuliers déterministes initialisés est PSPACE complet [65]. De plus, le problème de l’atteignabilité dans les automates temporisés définis par Alur et Dill est PSPACE complet s’ils sont déterministes et indécidables s’ils sont non déterministes [5]. Le même résultat a été prouvé pour des automates temporisés initialisés [65].

Ces résultats mènent à deux restrictions pour que le problème de l’atteignabilité dans les automates hybrides rectangulaires soit décidable. D’une part, les variables doivent être initialisées

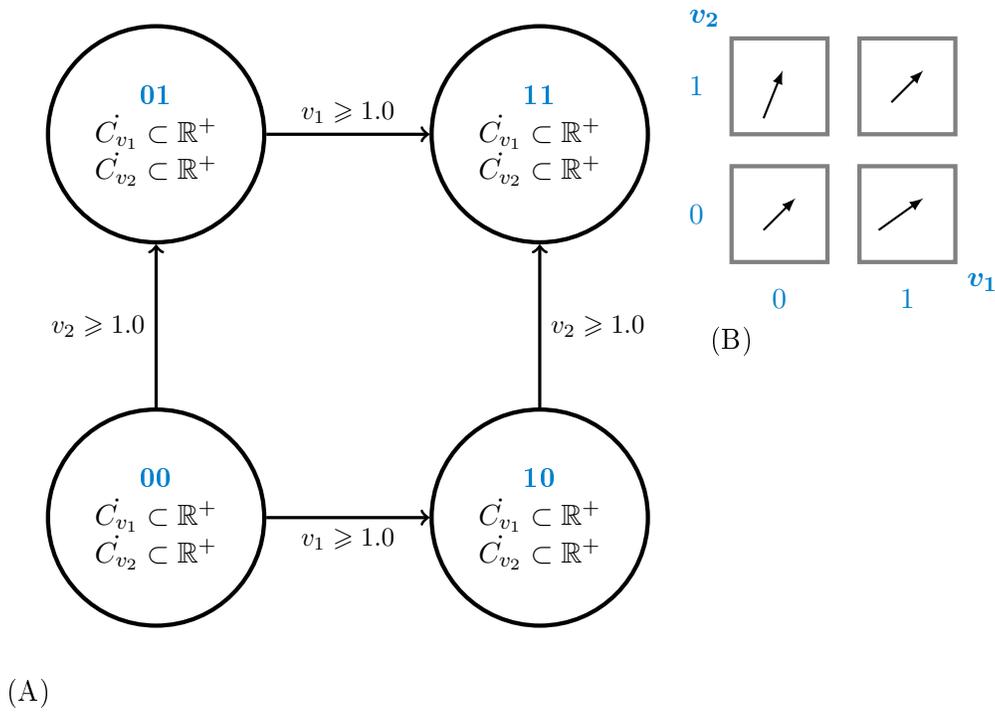


FIGURE 4.4 – Vue partielle d’un automate hybride AH non déterministe. (A) L’automate hybride est associé au graphe d’états présenté en (B). A partir de l’état qualitatif du système **00**, on peut atteindre le coin supérieur selon les flux identifiés et ainsi utiliser une transition discrète soit vers **01**, soit vers **10**. Ici encore, les glissements ne sont pas représentés pour des raisons pédagogiques.

et d’autre part, les invariants ne doivent pas comparer les variables entre elles.

L’automate hybride AH se rapproche de la définition d’un automate singulier non déterministe, voir figure 4.3 page précédente. En effet, les sauts ne sont pas déterministes : les flux dans un état mènent possiblement à atteindre plusieurs bords en même temps. Il est donc possible de satisfaire plusieurs gardes et donc d’avoir à choisir entre plusieurs transitions discrètes, voir figure 4.4. Le résultat de Henzinger sur la décidabilité de l’atteignabilité des automates singuliers déterministes ne s’applique donc pas. Il semble que le problème de l’atteignabilité pour les automates AH soit indécidable mais cela reste une conjecture. Cependant, dans le cas où toutes les célérités sont égales, l’automate hybride AH correspond à un automate temporisé dont le problème de l’atteignabilité est PSPACE complet.

Il faut toutefois prendre en compte que ces résultats s’intéressent uniquement à l’atteignabilité. Dans l’approche générale de modélisation des réseaux de régulation par AH, la première question porte sur la détermination des coefficients (flux) à partir des données expérimentales. Ces données d’entrée sont généralement la description d’une succession d’événements biologiques espacés par des durées mesurées. Cette connaissance partielle des traces de l’AH devrait

de fait réduire la complexité du problème.

Système dérivé constant par morceau

Les **systèmes dérivés constants par morceau** (PCD pour *Piecewise Constant Derivative system*) sont des automates hybrides linéaires dont la dynamique des variables x est déterminée par l'équation $\dot{x} = c$ avec c une constante [8]. Dans chaque état qualitatif du système, la transition continue évolue suivant un champ de vecteur constant. Les systèmes PCD sont alors non déterministes puisqu'il est possible de satisfaire plusieurs gardes en même temps contrairement aux automates singuliers. Une transition discrète est possible sur les frontières entre ces régions où les trajectoires changent de direction. La trajectoire représente donc des lignes brisées. Les automates AH sont similaires aux systèmes PCD, à la différence que AH permet des glissements le long des frontières et que les parties fractionnaires π sont réinitialisées après une transition discrète, voir figure 3.9 page 56.

La question d'atteignabilité est décidable pour les systèmes PCD à 2 dimensions, et est indécidable à plus haute dimension [7, 8]. En se comparant à cette classe d'automates, l'atteignabilité de l'approche AH semble être indécidable. On pourrait envisager de coder les automates AH en automates PCD quitte à augmenter le nombre de *locations*. Cette démarche pourrait permettre de prouver l'indécidabilité du problème de l'atteignabilité pour les automates AH lorsqu'il y a plus de 2 dimensions.

4.1.3 Les outils de vérification existants

Les outils de vérification des automates hybrides sont généralement spécifiques à une ou plusieurs classes d'automates hybrides. Parmi les nombreux outils de vérification d'automates hybrides existants, **HyTech** est un *model checker* symbolique vérifiant des propriétés de logiques temporelles [64]. C'est un outil adapté aux systèmes hybrides linéaires. HyTech réalise une exploration de l'espace d'états et si la vérification échoue, HyTech génère un contre-exemple montrant un cas où la propriété n'est pas satisfaite. Cet outil a montré son efficacité pour des automates hybrides [122].

Uppaal est un outil pour la modélisation, la simulation et la vérification de systèmes en temps réel tels que les *automates temporisés* [15, 84]. Les systèmes peuvent être modélisés par des processus non déterministes avec contrôle d'horloges à valeurs réelles. Uppaal contient un langage de description non déterministe, un outil de validation qui permet d'examiner l'exécution des dynamiques possibles avant de vérifier par *model checking* symbolique le comportement dynamique exhaustif du système. Pour faciliter la modélisation et la correction des programmes, le model-checker de Uppaal peut automatiquement générer un diagnostic de trace qui explique

pourquoi une propriété n'est pas satisfaite par un système. L'automate hybride AH n'étant pas un automate temporisé où toutes les célérités dans un état du système sont les mêmes, Uppaal n'est pas adaptable sans forte abstraction à tous les automates hybrides AH.

Roméo est un outil qui valide et vérifie par *model checking* les systèmes en temps réel représentés par des réseaux de Petri Temporels (TPNs) [51]. La vérification de propriétés complexes exprimées par l'utilisateur est possible dès lors que l'utilisateur traduit une propriété quantitative en test d'atteignabilité. Roméo peut aussi transformer ces réseaux en automates temporisés et ainsi leur permettre d'être vérifiés par d'autres outils de vérification tels que Uppaal. Les automates hybrides AH pourraient être partiellement vérifiés dans cet outil : puisqu'il est possible de transformer des systèmes discrets temporels en TPNs [32], on pourrait envisager de représenter la partie discrète des AH en TPNs en faisant l'abstraction de la dynamique dans chacun des états qualitatifs de l'automate.

Dans notre approche générale de modélisation de réseaux de régulation par automates hybrides, nous nous appuyons sur des observations biologiques partielles. Les outils de vérification des automates hybrides présentés ci-dessus semblent alors inadaptés. En effet, les connaissances biologiques fournies permettent d'utiliser une trajectoire connue pour identifier les flux de l'automate, autrement dit les paramètres dynamiques du système. Cependant, les observations biologiques ne permettent pas de mesurer l'ensemble des célérités. Les paramètres manquants doivent alors être instanciés sans connaissances précises sur eux, et l'énumération de tous les paramètres possibles est hors d'atteinte puisque ces paramètres sont dans \mathbb{R} . La solution proposée consiste donc à rechercher les contraintes sur les paramètres dynamiques qui rendent l'AH compatible avec les traces biologiques expérimentales. L'approche se base sur la **logique de Hoare**.

4.2 Logique de Hoare pour la vérification de modèles discrets

4.2.1 Vérification d'un programme

La **logique de Hoare** est une méthode formelle qui permet de prouver la correction de programmes. Elle a été défini par Tony Hoare en 1969 [67]. A chaque instruction syntaxique du programme correspond une règle de la logique de Hoare. Les instructions élémentaires du programme sont donc adossées aux règles d'inférence d'affectation, d'instruction conditionnelle, de composition de programme et d'itération (boucle *while*).

La logique de Hoare s'appuie sur la notion de **triplet de Hoare** noté $\{Pre\} p \{Post\}$ où *Pre* et *Post* sont des conditions d'états du système, et *p* est le programme. Un triplet est dit correct lorsque, en partant d'un état du système satisfaisant *Pre*, l'exécution du programme *p*

est possible, elle termine et mène à un état du système satisfaisant $Post$. Les règles d'inférence associées à l'affectation, l'instruction conditionnelle, la composition de programme et le *while* permettent de prouver efficacement qu'un programme, s'il termine, est correct. L'exemple le plus simple est celui où le programme est réduit à une incrémentation d'une variable x :

Exemple: Soit le triplet de Hoare $\{x = 0\} x := x + 1 \{x = 1\}$. Si la variable x prend la valeur 0 en précondition, le programme $x := x + 1$ est possible et mène à un état où la postcondition du triplet est satisfaite.

Edsger Dijkstra [40] a défini une sémantique de transformation des prédicats pour les langages de programmation impératifs : à chaque instruction est affectée une transformation de prédicats. A partir d'une instruction élémentaire IE est calculée "la plus faible précondition" de cette instruction qui correspond à une fonction associant à chaque postcondition $Post$ la plus faible précondition sur l'état initial assurant que l'exécution de IE termine dans un état final satisfaisant $Post$. Pour une succession d'instruction élémentaire $P; IE$ dont la dernière instruction est IE , c'est-à-dire pour tout programme séquentiel, et pour chaque postcondition $Post$, la transformation de prédicats de IE permet de déterminer la plus faible précondition avant la dernière instruction. En réitérant ce procédé, il est possible de déterminer la plus faible précondition du programme complet : cette méthode est appelée la **stratégie backward**.

Exemple: Dans le cas où deux entités échangent leurs valeurs (swap), le programme $p = \left\{ \begin{array}{l} aux := x; \\ x := y; \\ y := aux; \end{array} \right\}$, les pré et postconditions $Pre = \left\{ \begin{array}{l} x = x_{val} \\ y = y_{val} \end{array} \right\}$ et $Post = \left\{ \begin{array}{l} x = y_{val} \\ y = x_{val} \end{array} \right\}$ mènent au schéma de preuve suivant :

$$\begin{array}{c}
 \frac{Pre \Rightarrow \left\{ \begin{array}{l} x = x_{val} \\ y = y_{val} \end{array} \right\}}{\left\{ \begin{array}{l} x = x_{val} \\ y = y_{val} \end{array} \right\} aux := x \left\{ \begin{array}{l} aux = x_{val} \\ y = y_{val} \end{array} \right\} \quad \left\{ \begin{array}{l} aux = x_{val} \\ y = y_{val} \end{array} \right\} x := y \left\{ \begin{array}{l} x = y_{val} \\ aux = x_{val} \end{array} \right\}}{\left\{ Pre \right\} aux := x; x := y \left\{ \begin{array}{l} x = y_{val} \\ aux = x_{val} \end{array} \right\} \quad \left\{ \begin{array}{l} x = y_{val} \\ aux = x_{val} \end{array} \right\} y := aux \left\{ Post \right\}} \\
 \left\{ Pre \right\} p \left\{ Post \right\}
 \end{array}$$

La logique de Hoare a déjà été étendue pour s'adapter à des systèmes en temps réel tels que les communications asynchrones en incluant la composition parallèle [69]. Dans ce chapitre, nous définissons une nouvelle adaptation de la logique de Hoare pour la modélisation des comportements biologiques : le programme p est remplacé par une **trace biologique** qui représente des observations expérimentales sur le comportement qualitatif des entités du système au cours du temps. Similairement à la méthode de Dijkstra, nous définissons un calcul de plus faible précondition dans l'objectif d'obtenir l'ensemble des contraintes minimales sur les paramètres

dynamiques du modèle qui lui permet d'être cohérent avec la spécification exprimée sous forme de traces.

4.2.2 Logique de Hoare génétiquement modifiée

Des précédents travaux ont défini une version modifiée de la logique de Hoare (appelée *logique de Hoare génétiquement modifiée*) bien adaptée au cadre de modélisation discret de René Thomas [17]. Les triplets de Hoare sont définis selon des propriétés d'états (langage d'assertion) et des transitions d'états (langage de spécification de trace). Les règles d'inférence applicables à ces triplets sont les règles d'incrément, de décrément, d'assertion, de quantificateurs existentiel et universel, d'affectation, de composition séquentielle, d'itération et enfin la règle associée au programme vide. La stratégie *backward* décrit par Dijkstra est utilisée de la même manière et permet de contraindre les paramètres dynamiques pour que le système soit compatible avec la spécification donnée sous forme de traces.

Pour que la logique puisse être d'une quelconque utilité, il est nécessaire que tout ce qui peut être prouvé grâce aux règles d'inférence soit juste du point de vue sémantique, c'est ce qu'on appelle la **correction** de la logique. La logique de Hoare génétiquement modifiée a été prouvée correcte. De manière générale, la correction d'un algorithme se prouve en assurant que le programme termine et qu'il mène à un résultat correspondant à la spécification. On parle de correction partielle. Or une boucle *while* dans le langage de spécification de trace peut ne jamais terminer. Mais ici, puisqu'on est dans un espace fini, on peut trouver automatiquement les invariants de boucles, puis prouver la terminaison des programmes. La correction est alors prouvée par induction sur l'arbre de preuve $\{P\} p \{Q\}$, c'est-à-dire que chaque règle d'inférence est prouvée correcte. Pour chaque règle d'inférence, il suffit de démontrer l'existence d'une transition d'un état η satisfaisant les règles décrites en précondition P vers un ensemble d'états E (car les quantificateurs mènent possiblement à plusieurs états), et que pour tout état η' appartenant à E , η' satisfait la postcondition Q .

La **complétude** de la logique de Hoare est la propriété d'une logique qui assure que toutes les déductions sémantiques sont prouvables avec les règles d'inférence de la logique : si un triplet de Hoare est satisfait dans un réseau de gènes, alors il existe un arbre de preuve en utilisant les règles d'inférence de la logique et dont la racine de l'arbre est le triplet de Hoare d'intérêt. L'idée de la preuve consiste, pour un triplet de Hoare $\{Pre\} p \{Post\}$, à calculer les plus faibles invariants de toutes les boucles *while* des règles d'inférence et à prouver que la stratégie *backward* du calcul de la plus faible précondition mène à une postcondition qui est une conséquence sémantique de la précondition Pre ($Pre \Rightarrow$ Plus faible précondition).

Dans la suite de ce chapitre, nous définissons la sémantique de la logique de Hoare dans

le cadre de modélisation hybride, nous prouvons sa correction et sa complétude sémantique (le calcul de la plus faible précondition calcule bien la plus faible précondition) puis nous développons le calcul de plus faible précondition sur l'exemple du cycle négatif couramment observé en biologie.

4.3 Logique de Hoare hybride

Nous avons vu précédemment qu'il est possible d'adapter la logique de Hoare pour coller aux exigences de la modélisation des réseaux de régulation dans un cadre discret. Nos travaux nous ont mené à concevoir une nouvelle version de la logique de Hoare dans le cadre de modélisation hybride. L'enjeu derrière cette méthode formelle modifiée est de pouvoir décrire "en toute simplicité" une trace biologique d'un réseau de gène hybride dont le comportement est décrit par une succession d'événements espacés par des durées souvent observables.

L'approche générale de la logique de Hoare est conservée dans cette **logique de Hoare hybride**. Les pré et postconditions du triplet $\{Pre\} p \{Post\}$ sont définies par le langage de propriétés (Définition 4.1 page suivante) tandis que le chemin p est la représentation d'une trace biologique exprimée dans le langage de chemins (Définition 4.5 page 74).

4.3.1 Langage de propriétés

Les conditions d'états du système (tels que Pre et $Post$) sont définies par le **langage de propriétés** \mathcal{L}_P . Chaque état est décrit par un couple (D, H) où D correspond à une condition discrète et H à une condition hybride. La **condition discrète** D décrit exclusivement des termes discrets d'un état du système. Plus précisément, D porte sur l'ensemble des niveaux qualitatifs des entités dans l'état hybride courant $h = (\eta, \pi)$. La **condition hybride** H décrit les contraintes dans l'état hybride h et permet de mélanger des contraintes sur des termes discrets et continus. Ces termes continus comprennent notamment les célérités des différentes entités dans l'état hybride h et leurs parties fractionnaires.

Langage de propriétés \mathcal{L}_P

Définition 4.1: Les *termes* du langage de propriétés \mathcal{L}_P sont inductivement définis comme suit :

- ◇ Un *terme discret* est une variable η_u avec $u \in V$, ou une constante de \mathbb{N} ;
- ◇ Un *terme continu* est une variable π_u ou π'_u avec $u \in V$, ou une célérité $C_{u,\omega,n}$ avec $u \in V$, $\omega \subset R^-(u)$ et $n \in \llbracket 0, b_u \rrbracket$, ou une constante de \mathbb{R} ;
- ◇ Les connecteurs $+$, $-$, \times et $/$ créent des nouveaux termes par composition, le dernier étant seulement défini pour les termes continus. Nous utilisons leurs sémantiques usuelles.

Les *atomes discrets* sont de la forme $n \square n'$ où n et n' sont des termes discrets et les *atomes continus* sont de la forme $f \square f'$ où f et f' sont des termes continus où \square représente les comparateurs classiques.

Les *conditions discrètes* sont définies par : $D ::= a_d \mid \neg D \mid D \wedge D \mid D \vee D$ où a_d est un atome discret.

Les *conditions hybrides* sont définies par : $H ::= a_d \mid a_c \mid \neg H \mid H \wedge H \mid H \vee H$ où a_d et a_c sont respectivement un atome discret et un atome continu.

Une *propriété* est un couple (D, H) formé par une condition discrète D et une condition hybride H . L'ensemble des couples (D, H) ainsi définis forme le *langage de propriétés* \mathcal{L}_P .

Un état hybride satisfait une propriété $\varphi = (D, H)$ si et seulement si D et H sont évaluées à vrai dans l'état hybride h , en utilisant la signification usuelle des connecteurs.

On notera $h \models \varphi$.

Pour décrire ce qui se passe dans un état qualitatif, on va appeler π le vecteur des parties fractionnaires du point d'entrée dans l'état qualitatif et π' le vecteur des parties fractionnaires du point de sortie. On peut alors décrire dans le langage de propriétés \mathcal{L}_P des formules portant sur les points d'entrée et de sortie.

La figure 4.5 page suivante décrit les différents termes discrets et continus dans un état qualitatif. Le couple (D, H) tel que $D \equiv (\eta_{v_1} = 0) \wedge (\eta_{v_2} = 1)$ et $H \equiv (\pi_{v_1} = 0.4) \wedge (\pi_{v_2} = 1.0) \wedge (\pi'_{v_1} = 1.0) \wedge (\pi'_{v_2} = 0.5) \wedge (\Delta t_1 = 5.0) \wedge (C_{v_1, \rho(\eta, v_1), 0} > 0) \wedge (C_{v_2, \rho(\eta, v_2), 1} < 0)$ est une propriété qui exprime que le point d'entrée est h_0 et que le point de sortie est h'_0 .

4.3.2 Langage de chemins

Le **langage de chemins** va permettre de représenter la trace biologique qui provient la plupart du temps d'expériences biologiques répétées. La trace va jouer le rôle du programme

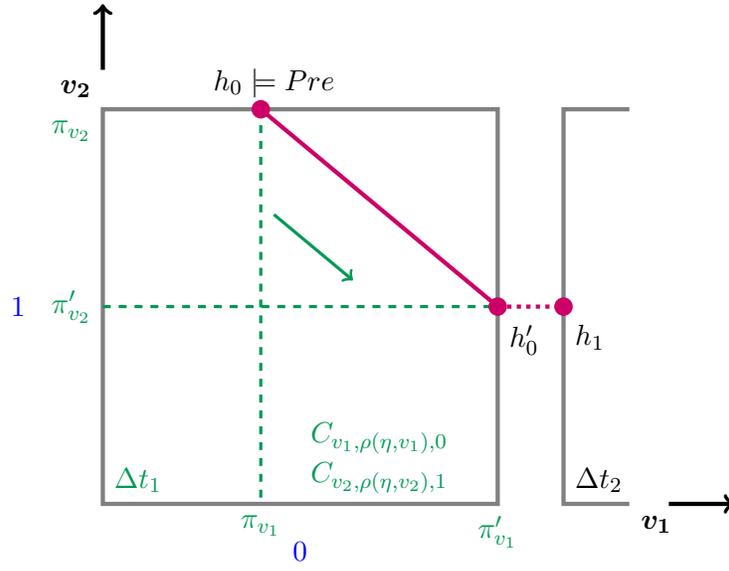


FIGURE 4.5 – Description des termes discrets et continus permettant de décrire la trajectoire à l'intérieur d'un état qualitatif. Les termes discrets sont représentés en bleu tandis que les termes continus sont représentés en vert. La transition continue de l'état qualitatif courant est représentée en trait plein rose tandis que la transition discrète est représentée en trait pointillé rose.

dans la logique de Hoare originelle, et permet de décrire un comportement biologique comme une succession d'événements qui arrivent à des instants précis. Le langage de chemins décrit alors ces traces biologiques : le chemin vide ϵ représente une observation vide d'événements, les **chemins élémentaires** de la forme $(\Delta t, a, dpa)$ représentent un événement isolé, et les chemins non élémentaires sont composés d'une séquence finie de chemins élémentaires, voir définition 4.5 page 74. L'objectif du langage de chemins est donc d'exprimer les transitions discrètes *via* les atomes de chemin discret (Définition 4.2 page suivante) et les transitions continues *via* le langage d'assertion \mathcal{L}_A (Définition 4.3 page suivante). Le temps passé pour effectuer la transition continue dans l'état qualitatif est représenté par Δt . La transition discrète, quand à elle, est instantanée.

Atome de chemin discret dpa .

Définition 4.2: Les *atomes de chemin discret* ou *dpa* (pour *discrete path atom*) sont définis par :

$$dpa ::= v+ \mid v-$$

où $v \in V$ est le nom d'une entité.

Pour tous les états $h = (\eta, \pi)$ et $h' = (\eta', \pi')$, la transition $h \xrightarrow{v+} h'$ (resp. $h \xrightarrow{v-} h'$) est satisfaite si et seulement si il existe une transition discrète de h vers h' tel que $\eta'_v = \eta_v + 1$ (resp. $\eta'_v = \eta_v - 1$).

Le prochain événement qualitatif est décrit par l'**atome de chemin discret** noté *dpa* (Définition 4.2). La prochaine transition discrète est connue puisque l'entité v exprimée dans le *dpa* franchit son bord supérieur (resp. inférieur) lorsque le symbole du dpa est le signe + (resp. -) : le niveau qualitatif η_v de l'entité v est incrémenté (resp. décrémente) de 1. Cette transition discrète a lieu uniquement lorsque le bord de l'entité est atteint. La figure 4.5 page précédente montre une transition discrète possible entre deux états qualitatifs (trait pointillé en rose) pour lequel le *dpa* correspond à v_1+ .

Langage d'assertion \mathcal{L}_A

Définition 4.3: Le langage d'assertion \mathcal{L}_A est défini par la grammaire suivante :

$$a ::= \top \mid C_v \square c \mid \text{slide}(v) \mid \text{slide}^+(v) \mid \text{slide}^-(v) \\ \mid \text{noslide}(v) \mid \text{noslide}^+(v) \mid \text{noslide}^-(v) \mid \neg a \mid a \wedge a \mid a \vee a$$

où $v \in V$ est le nom d'une entité, \square représente les comparateurs classiques et $c \in \mathbb{R}$ est un nombre réel.

Un couple $(\Delta t, a) \in \mathbb{R}^+ \times \mathcal{L}_A$ composé d'un nombre réel non-négatif et d'un élément du langage d'assertion est appelé un **couple d'assertion**.

Les comportements observés avant la prochaine transition discrète sont représentés par le langage d'assertion \mathcal{L}_A (Définition 4.3). L'assertion a traite d'informations additionnelles sur la transition continue fournies par l'utilisateur, et les symboles d'assertions spécifient des propriétés relatives à la dynamique dans l'état qualitatif courant :

- ◇ Le terme $C_v \square c$ contraint la célérité de l'entité v dans l'état qualitatif courant à être comparé à la constante c selon le comparateur \square .
- ◇ Le terme $\text{slide}^+(u)$ (resp. $\text{slide}^-(u)$) exprime la saturation (resp. la dégradation totale) de l'entité u : u atteint son bord supérieur (resp. inférieur) avant que l'entité v du *dpa*

atteigne le sien.

- ◇ A l'inverse, le terme $\text{noslide}^+(u)$ (resp. $\text{noslide}^-(u)$) empêche u d'atteindre son bord supérieur (resp. inférieur) avant que l'entité v du dpa atteigne le sien.
- ◇ Le terme $\text{slide}(u)$ (resp. $\text{noslide}(u)$) désigne la disjonction (resp. conjonction) de $\text{slide}^+(u)$ et $\text{slide}^-(u)$ (resp. $\text{noslide}^+(u)$ et $\text{noslide}^-(u)$).

En reprenant l'exemple de la figure 4.5 page 71, une assertion possible de la transition continue de l'état qualitatif courant observée en trait plein rose pourrait être $a \equiv \text{noslide}(v_2)$ puisque l'entité v_2 ne glisse pas sur un de ses bords avant que v_1 n'atteigne le sien.

Bien évidemment, il faut faire attention à ne pas dire dans un chemin élémentaire à la fois que c'est l'entité v qui passe son bord ($dpa \equiv v+$ ou $v-$) et que l'entité v glisse ou ne glisse pas ($a \equiv \text{slide}(v)$ ou $\text{noslide}(v)$). Notons aussi que lorsque l'utilisateur décrit la trace biologique, il indique l'assertion $a \equiv \text{True}$ s'il ne connaît pas d'informations supplémentaires.

Sémantique du couple d'assertion $(\Delta t, a)$

Définition 4.4: Considérons un état hybride $h = (\eta, \pi)$ et l'unique transition continue démarrant de h et finissant dans $h' = (\eta, \pi')$. La relation de satisfaction entre la transition continue $h \longrightarrow h'$ et un couple d'assertion $(\Delta t, a) \in \mathbb{R}^+ \times \mathcal{L}_A$ est notée $(h, h') \models (\Delta t, a)$, et est définie comme suit :

- ◇ Si $a \equiv \top$, $(h, h') \models (\Delta t, a)$ si et seulement si $\delta_h^{\text{first}} = \Delta t$.
- ◇ Si a est de la forme $(C_u \square c)$, $(h, h') \models (\Delta t, a)$ si et seulement si $\delta_h^{\text{first}} = \Delta t$ et $(C_{u, \rho(h, u), \eta_u} \square c)$.
- ◇ Si a est de la forme $\text{slide}(v)$, $(h, h') \models (\Delta t, a)$ ssi $\delta_h^{\text{first}} = \Delta t$ et $\delta_h(v) < \delta_h^{\text{first}}$.
- ◇ Si a est de la forme $\text{slide}^+(v)$ (resp. $\text{slide}^-(v)$), $(h, h') \models (\Delta t, a)$ ssi $\delta_h^{\text{first}} = \Delta t$ et $\delta_h(v) < \delta_h^{\text{first}}$ et $C_{v, \rho(h, v), \eta_v} > 0$ (resp. $C_{v, \rho(h, v), \eta_v} < 0$).
- ◇ Si a est de la forme $\text{noslide}(v)$, $(h, h') \models (\Delta t, a)$ ssi $\delta_h^{\text{first}} = \Delta t$ et $\delta_h(v) \geq \delta_h^{\text{first}}$.
- ◇ Si a est de la forme $\text{noslide}^+(v)$ (resp. $\text{noslide}^-(v)$), $(h, h') \models (\Delta t, a)$ ssi $\delta_h^{\text{first}} = \Delta t$ et $(C_{v, \rho(h, v), \eta_v} > 0) \Rightarrow (\delta_h(v) \geq \delta_h^{\text{first}})$ (resp. $(C_{v, \rho(h, v), \eta_v} < 0) \Rightarrow (\delta_h(v) \geq \delta_h^{\text{first}})$).
- ◇ Si a est de la forme $\neg a'$, $(h, h') \models (\Delta t, a)$ si et seulement si $\delta_h^{\text{first}} = \Delta t$ et $(h, h') \not\models (\Delta t, a')$.
- ◇ Si a est de la forme $a' \wedge a''$ (resp. $a' \vee a''$), $(h, h') \models (\Delta t, a)$ si et seulement si $(h, h') \models (\Delta t, a')$ et (resp. ou) $(h, h') \models (\Delta t, a'')$.

Une transition continue $h \longrightarrow h'$ satisfait donc le **couple d'assertion** $(\Delta t, a)$ si la transition existe, dure Δt unité de temps, et si elle respecte l'assertion a , voir définition 4.4 : \top est toujours vrai, $C_v \square c$ est satisfait si la célérité de v dans l'état qualitatif courant $C_{v, \rho(h, v), \eta_v}$ satisfait la relation $C_{v, \rho(h, v), \eta_v} \square c$, $\text{slide}^+(v)$ (resp. $\text{slide}^-(v)$) est satisfait si v atteint son bord

supérieur (resp. inférieur) et fait face à un mur, $\text{slide}(v)$ est satisfait si v atteint un de ses bords et fait face à un mur, $\text{noslides}^+(v)$ (resp. $\text{noslides}^-(v)$) est satisfait si v n'atteint pas son bord supérieur (resp. inférieur), $\text{noslides}(v)$ est satisfait si aucun des bords de v est atteint, et les connecteurs logiques ont leur signification usuelle.

Remarquons aussi que, d'après cette définition, $\text{slide}(v) \equiv \text{slide}^+(v) \vee \text{slide}^-(v)$ puisque, si $C_{v,\rho(h,v),\eta_v} = 0$, $\delta_h(v) = +\infty$ donc $\delta_h(v) < \delta_h^{\text{first}}$ est impossible. Ainsi, $\text{slide}(v) \equiv (\delta_h^{\text{first}} = \Delta t) \wedge (\delta_h(v) < \delta_h^{\text{first}}) \wedge (C_{v,\rho(h,v),\eta_v} > 0 \vee C_{v,\rho(h,v),\eta_v} < 0)$ est sémantiquement équivalent à $(\delta_h^{\text{first}} = \Delta t) \wedge (\delta_h(v) < \delta_h^{\text{first}})$.

Langage de chemins \mathcal{L}_C

Définition 4.5: Les *chemins (discrets)* sont définis par :

$$p ::= \varepsilon \mid (\Delta t, a, v\pm) \mid p ; p$$

où $(\Delta t, a)$ est un couple d'assertion et $v\pm$ est un atome de chemin discret. La sémantique d'un chemin p est donné par la relation binaire \xrightarrow{p} entre des états définis par :

- ◇ Si $p = \varepsilon$, alors $h_1 \xrightarrow{p} h_2$ si et seulement si $h_1 = h_2$;
- ◇ Si $p = (\Delta t, a, v\pm)$, alors $h_1 \xrightarrow{p} h_2$ si et seulement si il existe un état h'_1 tel que $h_1 \xrightarrow{(\Delta t, a)} h'_1$ et $h'_1 \xrightarrow{v\pm} h_2$;
- ◇ Si $p \equiv p_1 ; p_2$, alors $h_1 \xrightarrow{p} h_2$ si et seulement si il existe un état h_3 tel que $h_1 \xrightarrow{p_1} h_3$ et $h_3 \xrightarrow{p_2} h_2$.

Un chemin contenant uniquement $(\Delta t, a, v\pm)$ est appelé un *chemin élémentaire*.

En reprenant l'exemple de la figure 4.5 page 71, le chemin élémentaire $(5.0, \text{noslides}(v_2), v_1+)$ permet de définir la succession de la transition continue et de la transition discrète observée allant de h_0 à h_1 .

4.3.3 Triplets de Hoare et leur sémantique

A partir du langage de propriétés et du langage de chemins décrits précédemment, nous définissons les triplets de Hoare permettant de représenter des comportements biologiques observés expérimentalement, voir définition 4.6 page suivante.

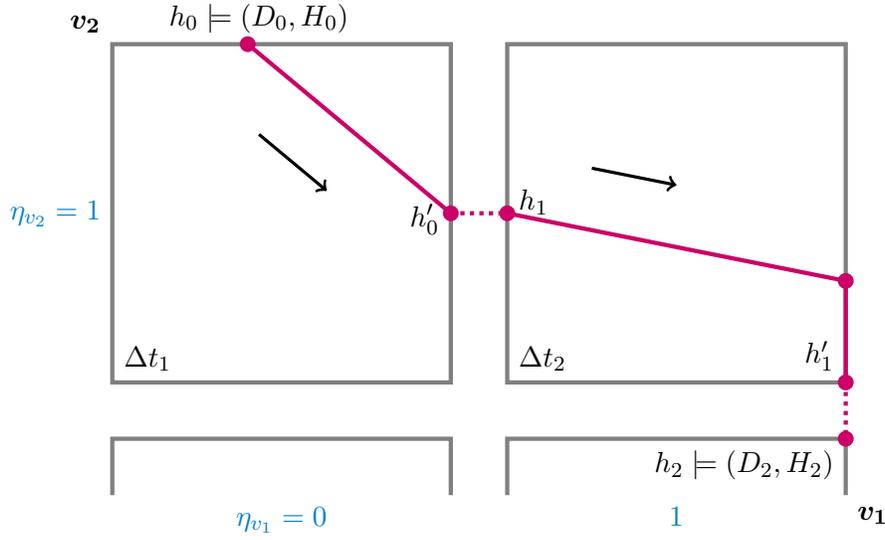


FIGURE 4.6 – Illustration d’une trace satisfaisant le triplet de Hoare : $\{D_0, H_0\} p \{D_2, H_2\}$ avec $p \equiv (\Delta t_1, \text{noslide}(v_2), v_1+); (\Delta t_2, \text{slide}^+(v_1), v_2-)$. A partir de l’état hybride $h_0 \models (D_0, H_0)$, une transition continue $h_0 \rightarrow h'_0$ d’une durée Δt_1 est possible suivie d’une transition discrète $h'_0 \rightarrow h_1$. Une nouvelle transition continue $h_1 \rightarrow h'_1$ d’une durée Δt_2 est possible et montre un glissement de v_1 sur son bord supérieur. Enfin, une transition discrète $h'_1 \rightarrow h_2$ a lieu, menant à $h_2 \models (D_2, H_2)$.

Triplet de Hoare hybride

Définition 4.6: Un triplet de Hoare hybride pour un réseau de gène hybride donné est une expression de la forme $\{Pre\} p \{Post\}$ où Pre et $Post$, appelés *précondition* et *postcondition* respectivement, sont des propriétés de \mathcal{L}_P , et p est un chemin de \mathcal{L}_C . Un triplet de Hoare $\{Pre\} p \{Post\}$ est *satisfait* si et seulement si pour tout état $h_1 \models Pre$, il existe un autre état h_2 tel que $h_1 \xrightarrow{p} h_2$ et $h_2 \models Post$.

La figure 4.6 illustre le triplet de Hoare suivant en matérialisant des transitions continues et discrètes possibles pour passer de l’état initial h_0 à l’état final h_2 :

$$\left\{ \begin{array}{l} D_0 \\ H_0 \end{array} \right\} \left(\begin{array}{c} \Delta t_1 \\ \text{noslide}(v_2) \\ v_1+ \end{array} \right); \left(\begin{array}{c} \Delta t_2 \\ \text{slide}^+(v_1) \\ v_2- \end{array} \right) \left\{ \begin{array}{l} D_2 \\ H_2 \end{array} \right\}$$

A partir d’un état hybride h_0 satisfaisant la précondition définie par le couple (D_0, H_0) , les transitions continue et discrète du chemin élémentaire $\left(\begin{array}{c} \Delta t_1 \\ \text{noslide}(v_2) \\ v_1+ \end{array} \right)$ mène à un état hybride h_1

en un temps Δt_1 , puis le second chemin élémentaire est matérialisé par une transition continue et une transition discrète menant à un état h_2 en un temps Δt_2 . Cet état hybride h_2 satisfait la postcondition définie par la propriété (D_2, H_2) .

Il est maintenant possible de définir un nouveau calcul de plus faible précondition, similaire à celui de Dijkstra, dans l'objectif de construire les contraintes minimales qui doivent satisfaire les paramètres dynamiques pour que le modèle soit en accord avec la trace biologique observée.

4.3.4 Calcul de la plus faible précondition

Rappelons que la méthode de Dijkstra s'appuie sur la transformation de prédicats. Dans notre formalisme, chaque chemin élémentaire de la trace biologique, noté ce , correspond à une transition continue d'une durée Δt , suivi d'une transition discrète instantanée. La **plus faible précondition** d'une trace biologique est la propriété (D, H) représentant les contraintes minimales que doivent satisfaire les paramètres cinétiques pour qu'il existe une trace illustrant le chemin spécifié.

Avant de pouvoir donner la plus faible précondition, il est nécessaire d'introduire quelques notions et notations. Puisque l'ensemble des ressources des entités va jouer un grand rôle, nous définissons la formule Φ_v^ω , pour toute entité $v \in V$ et pour tout sous-ensemble prédécesseur de v , $\omega \subset R^-(v)$:

$$\Phi_v^\omega \equiv \left(\bigwedge_{m \in \omega} \varphi_m \right) \wedge \left(\bigwedge_{m \in R^{-1}(v) \setminus \omega} \neg \varphi_m \right)$$

qui est vraie dans un état h si et seulement si les ressources de v en h sont exactement ω .

Nous introduisons maintenant 5 sous-propriétés :

1 - Transition discrète $\Phi_v^+(\Delta t)$. La sous-propriété $\Phi_v^+(\Delta t)$ (resp. $\Phi_v^-(\Delta t)$) donne les conditions assurant l'entité du *dpa* $v \in V$ d'incrémenter (resp. de décrémenter) son niveau qualitatif discret après Δt unités de temps :

- ◇ sa position de sortie¹ π'_v est localisée sur un bord ($\pi'_v = 1$ si le dpa est $v+$, $\pi'_v = 0$ si le dpa est $v-$),
- ◇ sa célérité dans l'état courant est positive (resp. négative),
- ◇ sa partie fractionnaire d'entrée dépend de sa partie fractionnaire de sortie, de Δt et de $C_{v,\rho(h,v),n}$. En d'autres termes, le délai de contact de v $\frac{\pi'_v - \pi_v}{C_{v,\omega,n}}$ correspond au temps Δt défini dans le chemin élémentaire

1. Dans la suite du manuscrit, lorsque les états hybrides ne sont pas spécifiés, nous notons π_v pour désigner la position d'entrée de v dans un état qualitatif, π'_v pour désigner sa position de sortie.

$$\Phi_v^+(\Delta t) \equiv (\pi'_v = 1) \wedge \bigwedge_{\substack{\omega \subset R^-(v) \\ n \in \llbracket 0, b_v \rrbracket}} \left((\Phi_v^\omega \wedge (\eta_v = n)) \Rightarrow \left(\begin{array}{l} (C_{v,\omega,n} > 0) \wedge \\ (\pi_v = \pi'_v - C_{v,\omega,n} \cdot \Delta t) \end{array} \right) \right)$$

$$\Phi_v^-(\Delta t) \equiv (\pi'_v = 0) \wedge \bigwedge_{\substack{\omega \subset R^-(v) \\ n \in \llbracket 0, b_v \rrbracket}} \left((\Phi_v^\omega \wedge (\eta_v = n)) \Rightarrow \left(\begin{array}{l} (C_{v,\omega,n} < 0) \wedge \\ (\pi_v = \pi'_v - C_{v,\omega,n} \cdot \Delta t) \end{array} \right) \right)$$

Les conjonctions indicées par ω et n permettent de prendre en compte le bon ensemble de ressources et le bon niveau qualitatif de v .

2 - Murs internes et externes \mathcal{W}_u^\pm . Pour toute entité $u \in V$, la sous-propriété \mathcal{W}_u^+ (resp. \mathcal{W}_u^-) empêche u d'incrémenter (resp. décrémenter) son niveau qualitatif de u . Ce mur peut être externe EW_u^+ (resp. EW_u^-) ou interne IW_u^+ (resp. IW_u^-).

$$\mathcal{W}_u^+ \equiv \text{EW}_u^+ \vee \text{IW}_u^+ \quad \text{et} \quad \mathcal{W}_u^- \equiv \text{EW}_u^- \vee \text{IW}_u^-$$

Les sous-formules $\text{EW}^\pm(u)$ expriment un mur externe pour l'entité u . La sous-formule $\text{EW}^+(u)$ (resp. $\text{EW}^-(u)$) requiert donc d'être au niveau qualitatif maximal $\eta_u = b_u$ (resp. minimal $\eta_u = 0$) pour l'entité u et que la célérité de u dans l'état courant soit positive (resp. négative) :

$$\text{EW}_u^+ \equiv (\eta_u = b_u) \wedge \bigwedge_{\omega \subset R^-(u)} (\Phi_u^\omega \Rightarrow C_{u,\omega,b_u} > 0)$$

$$\text{EW}_u^- \equiv (\eta_u = 0) \wedge \bigwedge_{\omega \subset R^-(u)} (\Phi_u^\omega \Rightarrow C_{u,\omega,0} < 0)$$

Les sous-formules $\text{IW}^\pm(u)$ décrivent les murs internes de l'entité u dans l'état courant ($\eta_u > 0$ ou $\eta_u < b_u$). $\text{IW}^+(u)$ (resp. $\text{IW}^-(u)$) indique que la célérité de l'entité u est positive (resp. négative) dans l'état courant tandis que sa célérité est négative (resp. positive) dans l'état voisin dans lequel le niveau qualitatif de u a été incrémenté (resp. décrémenté) de 1. Notons que, $\Phi_{u+}^{\omega'}$ (resp. $\Phi_{u-}^{\omega'}$), requis dans ces sous-formules, est vraie si et seulement si les ressources de u sont exactement ω' dans l'état où le niveau qualitatif de u a été incrémenté (resp. décrémenté) de 1.

$$\Phi_{u+}^{\omega'} \equiv (\eta_u < b_u) \wedge \bigwedge_{n \in \llbracket 0, b_u - 1 \rrbracket} ((\eta_u = n) \Rightarrow \Phi_u^{\omega'}[\eta_u \setminus \eta_u + 1])$$

$$\Phi_{u-}^{\omega'} \equiv (\eta_u > 0) \wedge \bigwedge_{n \in \llbracket 1, b_u \rrbracket} ((\eta_u = n) \Rightarrow \Phi_u^{\omega'}[\eta_u \setminus \eta_u - 1])$$

On peut maintenant donner une expression des formules IW_u^\pm :

$$\begin{aligned} \text{IW}_u^+ &\equiv (\eta_u < b_u) \wedge \bigwedge_{\substack{\omega, \omega' \subset R^-(u) \\ n \in \llbracket 0, b_u - 1 \rrbracket}} \left(\left(\begin{array}{l} (\eta_u = n) \wedge \\ (m = n + 1) \wedge \\ \Phi_u^\omega \wedge \Phi_{u+}^{\omega'} \end{array} \right) \Rightarrow \left(\begin{array}{l} C_{u, \omega, n} > 0 \wedge \\ C_{u, \omega', m} < 0 \end{array} \right) \right) \\ \text{IW}_u^- &\equiv (\eta_u > 0) \wedge \bigwedge_{\substack{\omega, \omega' \subset R^-(u) \\ n \in \llbracket 1, b_u \rrbracket}} \left(\left(\begin{array}{l} (\eta_u = n) \wedge \\ (m = n - 1) \wedge \\ \Phi_u^\omega \wedge \Phi_{u-}^{\omega'} \end{array} \right) \Rightarrow \left(\begin{array}{l} C_{u, \omega, n} < 0 \wedge \\ C_{u, \omega', m} > 0 \end{array} \right) \right) \end{aligned}$$

3 - Entités de *first* $\mathcal{F}(\Delta t)$. $\mathcal{F}(\Delta t)$ contraint les entités qui ne sont pas dans *first* à atteindre leur bord après les entités de *first*, ou à faire face à un mur interne ou externe si leur bord est atteint $((C_{u, \omega, n} > 0) \wedge (\pi_u > \pi'_u - C_{u, \omega, n} \cdot \Delta t))$ ou $(C_{u, \omega, n} < 0) \wedge (\pi_u < \pi'_u - C_{u, \omega, n} \cdot \Delta t)$.

$$\begin{aligned} \mathcal{F}(\Delta t) &\equiv \bigwedge_{u \in V \setminus \text{first}(h)} \left(\bigwedge_{\substack{\omega \subset R^-(u) \\ n \in \llbracket 0, b_u \rrbracket}} \left(\begin{array}{l} (\eta_u = n) \wedge \Phi_u^\omega \wedge \\ C_{u, \omega, n} > 0 \wedge \\ \pi_u > \pi'_u - C_{u, \omega, n} \cdot \Delta t \end{array} \right) \Rightarrow \mathcal{W}_u^+ \right) \\ &\quad \wedge \left(\bigwedge_{\substack{\omega \subset R^-(u) \\ n \in \llbracket 0, b_u \rrbracket}} \left(\begin{array}{l} (\eta_u = n) \wedge \Phi_u^\omega \wedge \\ C_{u, \omega, n} < 0 \wedge \\ \pi_u < \pi'_u - C_{u, \omega, n} \cdot \Delta t \end{array} \right) \Rightarrow \mathcal{W}_u^- \right) \end{aligned}$$

4 - Assertions hybrides $\mathcal{A}(\Delta t, a)$. La sous-propriété $\mathcal{A}(\Delta t, a)$ traduit les symboles d'assertion de la transition continue en contraintes :

$$\mathcal{A}(\Delta t, a) \equiv \bigwedge_{\substack{k \in \llbracket 1, n \rrbracket \\ \omega_k \in R^-(v_k) \\ n_k \in \llbracket 0, b_{v_k} \rrbracket}} \left(\bigwedge_{l \in \llbracket 1, n \rrbracket} \left((\eta_{v_l} = n_l) \wedge \Phi_{v_l}^{\omega_l} \right) \Rightarrow a \left[\begin{array}{l} C_{v_l} \setminus C_{v_l, \omega_l, n_l} \\ \text{slide}(v_l) \setminus \mathcal{S}_{v_l, \omega_l, n_l}(\Delta t) \\ \text{slide}^\pm(v_l) \setminus \mathcal{S}_{v_l, \omega_l, n_l}^\pm(\Delta t) \\ \text{noslide}(v_l) \setminus \text{no}\mathcal{S}_{v_l, \omega_l, n_l}(\Delta t) \\ \text{noslide}^\pm(v_l) \setminus \text{no}\mathcal{S}_{v_l, \omega_l, n_l}^\pm(\Delta t) \end{array} \right] \right)$$

avec a l'assertion du chemin élémentaire $(\Delta t, a, v\pm)$, et pour toute entité $u \in V \setminus \{v\}$:

$$\begin{aligned}
 \mathcal{S}_{u,\omega,n}^+(\Delta t) &\equiv (\pi'_u = 1) \wedge (\pi_u > \pi'_u - C_{u,\omega,n} \cdot \Delta t) \\
 \mathcal{S}_{u,\omega,n}^-(\Delta t) &\equiv (\pi'_u = 0) \wedge (\pi_u < \pi'_u - C_{u,\omega,n} \cdot \Delta t) \\
 \mathcal{S}_{u,\omega,n}(\Delta t) &\equiv \mathcal{S}_{u,\omega,n}^+(\Delta t) \vee \mathcal{S}_{u,\omega,n}^-(\Delta t) \\
 \text{no}\mathcal{S}_{u,\omega,n}^+(\Delta t) &\equiv (\pi'_u < 1) \wedge (C_{u,\omega,n} > 0 \Rightarrow \Delta t \leq \frac{\pi'_u - \pi_u}{C_{u,\omega,n}}) \\
 \text{no}\mathcal{S}_{u,\omega,n}^-(\Delta t) &\equiv (\pi'_u > 0) \wedge (C_{u,\omega,n} < 0 \Rightarrow \Delta t \leq \frac{\pi'_u - \pi_u}{C_{u,\omega,n}}) \\
 \text{no}\mathcal{S}_{u,\omega,n}(\Delta t) &\equiv \text{no}\mathcal{S}_{u,\omega,n}^+(\Delta t) \wedge \text{no}\mathcal{S}_{u,\omega,n}^-(\Delta t)
 \end{aligned}$$

Les trois premières sous-formules apportent une information sur la position de sortie π'_u : si l'entité u touche son bord, $\pi'_u = 0$ ou bien $\pi'_u = 1$. De plus, le délai de contact de u $\frac{\pi'_u - \pi_u}{C_{u,\omega,n}}$ est comparé avec le temps passé dans l'état Δt : lorsque le délai de contact est inférieur à Δt , l'entité u glisse sur son bord ($\pi_u > \pi'_u - C_{u,\omega,n} \cdot \Delta t$) pour $\mathcal{S}_{u,\omega,n}^+(\Delta t)$ et ($\pi_u < \pi'_u - C_{u,\omega,n} \cdot \Delta t$) pour $\mathcal{S}_{u,\omega,n}^-(\Delta t)$. Notons que les sous-formules $\mathcal{S}_u(\Delta t)$ et $\mathcal{S}_u^\pm(\Delta t)$ ne contraignent pas le signe de la célérité de l'entité glissante, qui sera par ailleurs contrainte par la sous-propriété \mathcal{F} .

Les trois dernières sous-formules indiquent que la position de sortie de l'entité u ne peut pas être sur un bord ($\pi'_u > 0$ et/ou $\pi'_u < 1$). Lorsque la célérité de l'entité u est positive (resp. négative) dans le cas de $\text{no}\mathcal{S}_{u,\omega,n}^+(\Delta t)$ (resp. $\text{no}\mathcal{S}_{u,\omega,n}^-(\Delta t)$), le temps passé dans l'état est inférieur ou égale au délai de contact de u ($\Delta t \leq \frac{\pi'_u - \pi_u}{C_{u,\omega,n}}$).

5 - Jonctions \mathcal{J}_v La sous-propriété \mathcal{J}_v établit des relations entre les états d'entrée et de sortie d'un état qualitatif (**Jonctions continues $\mathcal{C}\mathcal{J}_v$**) et entre l'état de sortie d'un état qualitatif et l'état d'entrée dans le suivant (**Jonctions discrètes $\mathcal{D}\mathcal{J}_v$**). Pour distinguer l'état de précondition et l'état de postcondition mentionnés dans cette sous-propriété, nous indexons les parties fractionnaires de la manière suivante : i pour l'état initial satisfaisant la précondition et f pour l'état final satisfaisant la postcondition. Notons que les sous-proprietés précédentes sont toutes définies dans l'état de précondition indicé i et que, pour éviter une surcharge de notation, nous avons remplacé les parties fractionnaires π_v^i et $\pi_v^{i'}$ par π_v et π'_v respectivement. Ainsi, \mathcal{J}_v est défini comme suit :

$$\mathcal{J}_v \equiv \mathcal{D}\mathcal{J}_v \wedge \bigwedge_{u \in V} \mathcal{C}\mathcal{J}_u$$

où $\mathcal{D}\mathcal{J}_v$ et $\mathcal{C}\mathcal{J}_v$ sont les propriétés des jonctions discrètes et continues définies ci-dessous.

Jonctions continues. Pour toute entité $v \in V$, et pour une transition continue entre deux états hybrides $h = (\eta, \pi)$ et $h' = (\eta, \pi')$, $\mathcal{C}\mathcal{J}_v$ établit une relation entre les parties fractionnaires et la célérité de l'entité v . Si la partie fractionnaire de sortie de v est 0 ou 1, le signe de la célérité peut être déduit et le temps nécessaire à v pour atteindre son bord est plus faible que le temps passé dans l'état qualitatif courant. Si v n'atteint pas son bord, la position exacte de la partie fractionnaire d'entrée de v peut être déduite de la position de sortie, du temps passé dans l'état qualitatif courant et de la célérité.

$$\mathcal{C}\mathcal{J}_v \equiv \begin{cases} (0 < \pi'_v < 1) \Rightarrow (\pi_v = \pi'_v - C_{v,\rho(h,v),\eta_v} \cdot \delta_h^{\text{first}}) \\ \wedge (\pi'_v = 0) \Rightarrow C_{v,\rho(h,v),\eta_v} < 0 \wedge \\ \quad (\pi_v \leq \pi'_v - C_{v,\rho(h,v),\eta_v} \cdot \delta_h^{\text{first}}) \\ \wedge (\pi'_v = 1) \Rightarrow C_{v,\rho(h,v),\eta_v} > 0 \wedge \\ \quad (\pi_v \geq \pi'_v - C_{v,\rho(h,v),\eta_v} \cdot \delta_h^{\text{first}}) \end{cases}$$

Jonctions discrètes. Pour toute entité $v \in V$, et pour une transition discrète modifiant le niveau qualitatif de v entre un état initial et un état final correspondant aux indices i et f , $\mathcal{D}\mathcal{J}_v$ établit une jonction entre les parties fractionnaires de ces états. Cette formule montre que la partie fractionnaire de v change de 1 à 0 (resp. 0 à 1) lors d'une incrémentation (resp. décrémentation) du niveau qualitatif de v tandis que les parties fractionnaires des autres entités restent inchangées :

$$\mathcal{D}\mathcal{J}_v \equiv (\pi_v^f = 1 - \pi_v^{i'}) \wedge \bigwedge_{u \in V \setminus \{v\}} (\pi_u^f = \pi_u^{i'}) .$$

Les contraintes résultant de la jonction discrète sont facilement interprétables sur la transition discrète de la figure 4.5 page 71 : toutes les parties fractionnaires restent les mêmes entre les deux états qualitatifs sauf la partie fractionnaire de l'entité v_1 assurant la transition.

L'ensemble des ces sous-propriétés donne des contraintes entre les points d'entrée et sortie des états qualitatifs et les paramètres dynamiques du modèle. Ces propriétés permettent de définir le calcul de plus faible précondition :

Plus faible précondition

Définition 4.7: Soit p une trace biologique exprimée dans le langage de chemins et $Post = (D_f, H_f)$ une postcondition indexée par un indice final f . La *plus faible précondition* attribuée à p et $Post$ est une propriété : $WP_f^i(p, Post) \equiv (D_{i,f}, H_{i,f})$, indexée par un indice initial i et par f définie récursivement de la façon suivante :

- ◇ Si $p = \epsilon$, alors $D_{i,f} \equiv D_f$ et $H_{i,f} \equiv H_f$;
- ◇ Si $p = (\Delta t, a, v+)$, alors :

$$D_{i,f} \equiv D_f[\eta_v \setminus \eta_v + 1],$$

$$H_{i,f} \equiv H_f[\eta_v \setminus \eta_v + 1] \wedge \Phi_v^+(\Delta t) \wedge \mathcal{F}(\Delta t) \wedge \neg \mathcal{W}_v^+ \wedge \mathcal{A}(\Delta t, a) \wedge \mathcal{J}_v;$$

- ◇ Si $p = (\Delta t, a, v-)$, alors :

$$D_{i,f} \equiv D_f[\eta_v \setminus \eta_v - 1],$$

$$H_{i,f} \equiv H_f[\eta_v \setminus \eta_v - 1] \wedge \Phi_v^-(\Delta t) \wedge \mathcal{F}(\Delta t) \wedge \neg \mathcal{W}_v^- \wedge \mathcal{A}(\Delta t, a) \wedge \mathcal{J}_v;$$

- ◇ Si $p = p_1; p_2$ est une séquence de chemins :

$$WP_f^i(p_1; p_2, Post) \equiv WP_m^i(p_1, WP_f^m(p_2, Post))$$

indexée par un nouvel indice intermédiaire m ;

avec les sous-propriétés $\Phi_v^\pm(\Delta t)$, \mathcal{W}_v^\pm , $\mathcal{F}(\Delta t)$, $\mathcal{A}(\Delta t, a)$ et \mathcal{J}_v introduites précédemment.

Néanmoins, le calcul de plus faible précondition de la définition précédente ne donne pas un moyen direct de calculer cette plus faible précondition pour une trace biologique donnée. En effet, la définition récursive partage un chemin en deux, sans expliciter où le programme est coupé en deux sous-programmes. C'est là que la stratégie *backward* joue son rôle. Pour une trace biologique p et une postcondition $Post$, la stratégie *backward* permet de remonter le chemin pas à pas en partant de la postcondition :

- ◇ Si p est de la forme ϵ ou $(\Delta t, a, v\pm)$, alors nous calculons la précondition comme décrit en définition 4.7.
- ◇ Si $p = p_1; ce$ avec $ce = (\Delta t, a, v\pm)$ le dernier chemin élémentaire, alors nous calculons la précondition avant ce et nous réitérons la stratégie sur le chemin p_1 .

Les deux étapes de la stratégie sont mutuellement exclusives ce qui signifie que la stratégie de preuve génère un unique arbre de preuve.

Une implémentation du calcul de plus faible précondition avec la stratégie backward a été

réalisée² et intégrée dans le programme **Holmes BioNet** détaillé dans le chapitre 6 page 113.

4.3.5 Règles d'inférence

Les **règles d'inférence** sont des fonctions qui associent à un ensemble de formules appelées prémisses une autre formule nommée conclusion. Elles s'écrivent généralement sous la forme $\frac{Prémisse(s)}{Conclusion}$. La conclusion est dite déductible des prémisses, ce que l'on note $Prémisse(s) \vdash Conclusion$. Lorsque la prémisse est vide, la conclusion est appelée théorème ou axiome.

Dans le système formel de la logique de Hoare hybride, les formules manipulées sont des triplets de Hoare dont l'ensemble des symboles est défini à partir du langage de propriétés et du langage d'assertion. Trois règles d'inférence sont définies :

1 - Règle d'incrémentatation :

$$\frac{}{\left\{ \begin{array}{l} D_f[\eta_v \setminus \eta_v + 1] \\ H_{i,f} \end{array} \right\} \left(\begin{array}{c} \Delta t \\ a \\ v+ \end{array} \right) \left\{ \begin{array}{l} D_f \\ H_f \end{array} \right\}}$$

2 - Règle de décrémentatation :

$$\frac{}{\left\{ \begin{array}{l} D_f[\eta_v \setminus \eta_v - 1] \\ H_{i,f} \end{array} \right\} \left(\begin{array}{c} \Delta t \\ a \\ v- \end{array} \right) \left\{ \begin{array}{l} D_f \\ H_f \end{array} \right\}}$$

où v est une entité, η_v son niveau qualitatif, D_f la condition discrète, H_f la condition hybride, $H_{i,f} \equiv H_f[\eta_v \setminus \eta_v + 1] \wedge \Phi_v^+(\Delta t) \wedge \mathcal{F}(\Delta t) \wedge \neg \mathcal{W}_v^+ \wedge \mathcal{A}(\Delta t, a) \wedge \mathcal{J}_v$ (resp. $H_{i,f} \equiv H_f[\eta_v \setminus \eta_v - 1] \wedge \Phi_v^-(\Delta t) \wedge \mathcal{F}(\Delta t) \wedge \neg \mathcal{W}_v^- \wedge \mathcal{A}(\Delta t, a) \wedge \mathcal{J}_v$) la condition hybride d'incrémentatation (resp. de décrémentatation) détaillée en sous-section 4.3.4 page 76, Δt le temps passé dans l'état qualitatif courant et a une assertion.

3 - Règle de composition séquentielle : $\frac{\{Q_1\} p_1 \{Q_3\} \quad \{Q_3\} p_2 \{Q_2\}}{\{Q_1\} p_1; p_2 \{Q_2\}}$

avec Q_1, Q_2, Q_3 les propriétés de la forme (D, H) ayant le rôle de pré et postconditions, et p_1 et p_2 les chemins déduits des observations biologiques.

Trois axiomes complètent ces règles d'inférence :

- ◇ $(\eta_v \geq 0) \wedge (\eta_v \leq b_v)$: le niveau qualitatif de chaque entité doit être dans son domaine de définition,
- ◇ $\forall v \in V, \forall \omega \subset R^-(v), \forall k \in \llbracket 0, b_v - 1 \rrbracket, C_{v,\omega,k} \times C_{v,\omega,k+1} \geq 0$: pour deux états qualitatifs voisins, si l'entité v est contrôlée par la même ressource, alors les célérités de v ne peuvent être de signes opposés.

2. Programme disponible à l'url suivante : <http://www.i3s.unice.fr/~comet/DOCUMENTS/hybridisation.tar.gz>

$$\diamond \forall v \in V, \forall \omega \subset R^-(v), \forall n \in \llbracket 0, b_v \rrbracket, C_{v,\omega,n} = 0 \Rightarrow \begin{cases} \forall i \in \llbracket n+1, b_v \rrbracket & C_{v,\omega,i} < 0 \\ \forall i \in \llbracket 0, n-1 \rrbracket & C_{v,\omega,i} > 0 \end{cases}$$

Le second schéma d'axiomes impose que les célérités d'une entité v entre deux états voisins font tendre le système vers le même état d'équilibre lorsque ces célérités ont les mêmes ressources, c'est-à-dire que les entités sont sous l'influence des mêmes prédécesseurs actifs, voir figure 3.5 page 51.

Le troisième schéma d'axiomes contraint le signe des célérités d'une entité v lorsque ces célérités ont les mêmes ressources et que le système tend vers un état d'équilibre où la célérité vaut 0.

Les règles de quantificateurs existentiel ou universel, d'affectation, d'itération et de boucle *while* (présentes dans la logique de Hoare génétiquement modifiée) ne sont pas définies pour ce formalisme puisque les données d'entrée correspondent à un comportement biologique observé expérimentalement : à partir d'une trace, il est difficile d'extrapoler la trace pour en extraire un chemin contenant un *while*. A l'opposé, il se peut qu'à la suite d'un ensemble d'expériences, on souhaite exprimer l'existence d'un chemin permettant d'atteindre un état final ou même dire que quelque soit le chemin parmi un ensemble, le système aboutira à un état satisfaisant une certaine propriété. Dans la pratique, la trace observée ne comporte ni de quantificateur ni de boucle *while*.

Maintenant que les règles d'inférence ont été définies, nous allons montrer que cette logique est correcte (tout triplet de Hoare pour lequel il existe un arbre de preuve avec les règles d'inférence est correct sémantiquement) et complète sémantiquement (le calcul de la plus faible précondition calcule bien la plus faible précondition).

Théorème 1. *La logique de Hoare hybride est correcte.*

4.4 Preuve de correction

La preuve de correction de la logique de Hoare hybride consiste à prouver que s'il existe une preuve d'un triplet de Hoare (hybride) en utilisant des règles d'inférence ($\vdash \{Pre\} p \{Post\}$), alors il est sémantiquement correct. Autrement dit, pour tout état η satisfaisant Pre , il existe un état η' tel que $\eta \xrightarrow{p} \eta'$ et $\eta' \models Post$. Nous allons décomposer la preuve de correction en faisant une preuve pour chaque règle d'inférence et pour chaque symbole d'assertion du langage d'assertion \mathcal{L}_A . Nous présentons uniquement la preuve de correction de la règle d'incrémentement puisque la preuve de correction de la décrémentement est similaire, et que celle de la règle de composition séquentielle est classique. Dans cette preuve, nous considérons un état hybride h ainsi que la règle d'incrémentement définie en sous-section 4.3.5 page ci-contre. Au préalable, nous définissons deux lemmes qui seront utilisés dans la preuve de correction.

4.4.1 Deux lemmes

Le lemme 1 affirme que les délais de contact des entités dans *first* correspondent au temps passé dans l'état Δt .

Lemme 1 (Temps passé dans un état qualitatif). *Soit h un état hybride. Si $h \models (D_1, H_1)$ et $H_1 \Rightarrow \Phi_v^+(\Delta t) \wedge \mathcal{F}(\Delta t)$ (resp. $H_1 \Rightarrow \Phi_v^-(\Delta t) \wedge \mathcal{F}(\Delta t)$), alors : $\delta_h^{\text{first}} = \delta_h(v) = \Delta t$.*

Preuve. Considérons $H_1 \Rightarrow \Phi_v^+(\Delta t) \wedge \mathcal{F}(\Delta t)$, et $h = (\eta, \pi)$ un état hybride tel que $h \models (D_1, H_1)$. A partir de la sous-propriété $\Phi_v^+(\Delta t)$ de la sous-section 4.3.4 page 76 l'entité v atteint son bord supérieur ($\pi'_v = 1$) et sa célérité est positive ($C_{v,\omega,n} > 0$).

Soit $h' = (\eta, \pi')$ l'état hybride où v est la première entité à atteindre son bord. Le temps passé dans l'état qualitatif courant η correspond au temps nécessaire pour atteindre le bord où v change son niveau qualitatif. En effet, à partir de $\Phi_v^+(\Delta t)$, on peut déduire que :

$$\pi_v = \pi'_v - C_{v,\omega,n} \cdot \Delta t = 1 - C_{v,\omega,n} \cdot \Delta t \quad \text{et ainsi} \quad \Delta t = \frac{1 - \pi_v}{C_{v,\omega,n}}.$$

La définition 3.4 page 51 du délai de contact nous permet de dire que $\Delta t = \delta_h(v)$.

Finalement, la sous-propriété $\mathcal{F}(\Delta t)$ de la sous-section 4.3.4 page 78 indique que si une entité u différente de v atteint son bord avant v , alors u fait face à un mur interne ou externe. Ainsi, puisque $\delta_h(v) = \Delta t$, et d'après la définition 3.6 page 53, on peut déduire que $\delta_h^{\text{first}} = \delta_h(v) = \Delta t$. La preuve dans le cas où $H_1 \Rightarrow \Phi_v^-(\Delta t) \wedge \mathcal{F}(\Delta t)$ est similaire \square

Le lemme 2 porte sur la préservation de l'évaluation des formules du calcul de plus faible précondition le long d'une transition continue. Autrement dit, entre deux états hybrides $h = (\eta, \pi)$ et $h' = (\eta, \pi')$ tels que $h \rightarrow h'$, il n'y a pas de perte d'informations.

Lemme 2 (Préservation de l'évaluation des formules le long d'une transition continue). *Considérons un triplet de Hoare $\{(D_{i,f}, H_{i,f})\} p \{(D_f, H_f)\}$ obtenu par le calcul de plus faible précondition, un état hybride $h = (\eta, \pi^i)$, et l'unique transition continue $h \rightarrow h'$ démarrante de h . Si $h \models (D_{i,f}, H_{i,f})$, alors $h' \models (D_{i,f}, H_{i,f})$.*

Preuve.

- ◇ Puisque h et h' appartiennent au même état qualitatif, les niveaux qualitatifs de toutes les entités sont les mêmes. L'évaluation de $D_{i,f}$ dans h' est alors identique à l'évaluation de $D_{i,f}$ dans h .
- ◇ Les termes de $H_{i,f}$ sont construits sur des célérités ou sur des coordonnées discrètes (η) ou continues (π) des différents points de la trajectoire (points d'entrée et de sortie dans les différents états qualitatifs). Ces points correspondent aux points h ou h' , ou bien sont à l'extérieur de l'état qualitatif courant.

De plus, les célérités sont constantes, les points de la trajectoire qui n'appartiennent pas à l'état qualitatif courant ne changent pas, ainsi que les points h et h' .

L'interprétation de $(D_{i,f}, H_{i,f})$ est donc la même dans h et dans h' . \square

4.4.2 Preuve de la correction par induction

La preuve de la correction de chacune des règles d'inférence de la logique de Hoare hybride est faite, pour tous les symboles d'assertion, en trois étapes :

- α) la transition continue existe,
- β) la transition discrète existe,
- γ) la postcondition est satisfaite après le chemin élémentaire.

Nous considérons donc un chemin élémentaire $\left(\begin{smallmatrix} \Delta t \\ a \\ v+ \end{smallmatrix}\right)$, une postcondition (D_f, H_f) et la plus faible précondition $(D_{i,f}, H_{i,f})$ calculée par la règle d'inférence d'incréméntation.

Premier cas : $a = \top$

α) D'après la sous-propriété $\Phi_v^+(\Delta t)$, $\pi'_v = 1$ (atteinte du bord supérieur de v), $C_{v,\omega,n} > 0$ et $\pi'_v = \pi_v + C_{v,\omega,n} \times \Delta t$ (le temps passé dans l'état courant est Δt) si les ressources de v sont ω et le niveau qualitatif de v est n , autrement dit si Φ_v^ω et $\eta_v = n$ sont satisfaites.

Considérons l'unique état hybride $h' = (\eta, \pi')$ tel que la transition continue $h \rightarrow h'$ existe. Ainsi, d'après le lemme 1 page précédente et la sémantique du couple d'assertion (définition 4.4 page 73), $(h, h') \models (\Delta t, \top)$.

β) En simplifiant la sous-propriété $\neg \mathcal{W}_v^+ \equiv \neg \text{EW}_v^+ \wedge \neg \text{IW}_v^+$ satisfaite en h' , nous obtenons :

$$\neg \text{EW}_v^+ \equiv \neg \left[\underbrace{(\eta_v = b_v)}_{\perp} \wedge \bigwedge_{\omega \subset R^-(v)} (\Phi_v^\omega \Rightarrow C_{v,\omega,b_v} > 0) \right] \equiv \top$$

qui est évaluée à vrai puisque v augmente son niveau qualitatif ($v+$ est le *dpa* du chemin élémentaire) et ainsi n'est pas situé sur son niveau qualitatif maximal b_v . On obtient donc $\neg \mathcal{W}_v^+ \equiv \neg \text{IW}_v^+$:

$$\begin{aligned} \neg \mathcal{W}_v^+ &\equiv \neg \left[(\eta_v < b_v) \wedge \bigwedge_{\substack{n \in \llbracket 0, b_v \rrbracket \\ \omega, \omega' \subset R^-(v)}} \left(\begin{array}{l} (\eta_v = n) \wedge \\ (m = n + 1) \wedge \\ \Phi_v^\omega \wedge \Phi_{v+}^{\omega'} \end{array} \Rightarrow \begin{array}{l} C_{v,\omega,n} > 0 \\ \wedge \\ C_{v,\omega',m} < 0 \end{array} \right) \right] \\ &\equiv \underbrace{\neg(\eta_v < b_v)}_{\perp} \vee \bigvee_{\substack{n \in \llbracket 0, b_v \rrbracket \\ \omega, \omega' \subset R^-(v)}} \left(\begin{array}{l} (\eta_v = n) \wedge \\ (m = n + 1) \wedge \\ \Phi_v^\omega \wedge \Phi_{v+}^{\omega'} \end{array} \Rightarrow \begin{array}{l} C_{v,\omega,n} > 0 \\ \wedge \\ C_{v,\omega',m} < 0 \end{array} \right) \\ &\text{puisque } \eta_v < b_v \end{aligned}$$

Parmi toutes les prémisses des disjonctions restantes, seulement une est satisfaite car l'état qualitatif courant et l'état suivant ont chacun un unique état qualitatif ($\eta_v = n$ et $m = n + 1$) et un unique ensemble de ressources (ω et ω'). En nommant le prochain état qualitatif η'' et en remplaçant ω et ω' par les ressources des états correspondant $\rho(\eta, v)$ et $\rho(\eta'', v)$, on en déduit :

$$\neg \mathcal{W}_v^+ \equiv \neg \left(\underbrace{\neg \left[\begin{array}{c} (\eta_v = n) \wedge \\ (m = n + 1) \wedge \\ \Phi_v^{\rho(\eta, v)} \wedge \Phi_{v+}^{\rho(\eta'', v)} \end{array} \right]}_{\perp} \vee \left(\begin{array}{c} C_{v, \rho(\eta, v), \eta_v} > 0 \\ \wedge \\ C_{v, \rho(\eta'', v), \eta''_v} < 0 \end{array} \right) \right) \equiv \left(\begin{array}{c} C_{v, \rho(\eta, v), \eta_v} \leq 0 \\ \vee \\ C_{v, \rho(\eta'', v), \eta''_v} \geq 0 \end{array} \right)$$

Cependant, puisque $\Phi_v^+(\Delta t)$ est vrai en h' , nous obtenons $C_{v, \rho(\eta, v), \eta_v} > 0$. Ainsi, $\neg \mathcal{W}_v^+$ est équivalent à $C_{v, \rho(\eta'', v), \eta''_v} \geq 0$ et l'inégalité précédente est vraie puisque $\neg \mathcal{W}_v^+$ est satisfaite en h' . L'entité v atteint donc son bord en Δt unités de temps ($\Phi_v^+(\Delta t)$) et le franchit ($\neg \mathcal{W}_v^+$), autorisant une transition discrète $h' \rightarrow h''$ augmentant le niveau qualitatif de v car les signes des célérités de v en h' et en h'' sont les mêmes.

γ) Nous avons prouvé précédemment qu'il existe une unique transition continue $h \rightarrow h'$ et une transition discrète $h' \rightarrow h''$. Puisque $h \models (D_f[\eta_v \setminus \eta_v + 1], H_{i, f})$, on déduit d'après le lemme 2 page 84 : $h' \models (D_f[\eta_v \setminus \eta_v + 1], H_{i, f})$. La transition discrète augmente le niveau qualitatif de v ($\eta''_v = \eta_v + 1$), ce qui nous permet d'en déduire :

$$h'' \models (D_f[\eta_v \setminus \eta_v + 1][\eta_v \setminus \eta_v - 1], H_{i, f}[\eta_v \setminus \eta_v - 1]), \quad \text{et ainsi} \quad h'' \models (D_f, H_{i, f}[\eta_v \setminus \eta_v - 1])$$

La condition hybride $H'' \equiv H_{i, f}[\eta_v \setminus \eta_v - 1]$ est satisfaite en h'' :

$$\begin{aligned} H'' &\equiv \left(H_f[\eta_v \setminus \eta_v + 1] \wedge \Phi_v^+(\Delta t) \wedge \mathcal{F}(\Delta t) \wedge \neg \mathcal{W}_v^+ \wedge \mathcal{A}(\Delta t, a) \wedge \mathcal{J}_v \right) [\eta_v \setminus \eta_v - 1] \\ &\equiv H_f \wedge \left(\Phi_v^+(\Delta t) \wedge \mathcal{F}(\Delta t) \wedge \neg \mathcal{W}_v^+ \wedge \mathcal{A}(\Delta t, a) \wedge \mathcal{J}_v \right) [\eta_v \setminus \eta_v - 1] \end{aligned}$$

Pour rappel, les points d'entrée et de sortie ne sont pas "mis à jour" ici.

En conclusion de cette preuve, les conditions discrète et hybride D_f et H_f sont satisfaites en h'' et la postcondition est vérifiée.

Deuxième cas : $a = \text{slide}^+(u)$ ou $a = \text{slide}^-(u)$ avec $u \neq v$

Nous ne traitons que avec le cas slide^+ , le cas slide^- étant similaire.

α) Similairement au premier cas, nous considérons l'unique état hybride $h' = (\eta, \pi')$ tel que la transition continue $h \rightarrow h'$ existe. Le temps passé dans l'état qualitatif courant est aussi Δt

(sous-propriété $\Phi_v^+(\Delta t)$). Puisque $a \neq \top$, la sous-propriété $\mathcal{A}(\Delta t, a)$ joue un rôle important :

$$\mathcal{A}(\Delta t, a) \equiv \bigwedge_{\substack{k \in \llbracket 1, n \rrbracket \\ \omega_k \in R^-(v_k) \\ n_k \in \llbracket 0, b_{v_k} \rrbracket}} \left(\bigwedge_{l \in \llbracket 1, n \rrbracket} \left((\eta_{v_l} = n_l) \wedge \Phi_{v_l}^{\omega_l} \right) \Rightarrow a \left[\begin{array}{c} C_{v_l} \setminus C_{v_l, \omega_l, n_l} \\ \text{slide}(v_l) \setminus \mathcal{S}_{v_l, \omega_l, n_l}(\Delta t) \\ \text{slide}^+(v_l) \setminus \mathcal{S}_{v_l, \omega_l, n_l}^+(\Delta t) \\ \text{slide}^-(v_l) \setminus \mathcal{S}_{v_l, \omega_l, n_l}^-(\Delta t) \end{array} \right] \right)$$

Parmi toutes les prémisses de ces conjonctions, seulement une est satisfaite car l'état qualitatif courant a un unique niveau qualitatif pour chaque entité v_l ($\eta_{v_l} = n_l$) et un unique ensemble de ressources pour chaque v_l ($\Phi_{v_l}^{\omega_l}$). Nous pouvons alors remplacer $\text{slide}^+(u)$ par la sous-formule \mathcal{S}^+ :

$$\mathcal{S}_{u, \omega, n}^+(\Delta t) \equiv (\pi_u' = 1) \wedge (\pi_u > \pi_u' - C_{u, \omega, n} \cdot \Delta t)$$

où ω correspond à l'ensemble des ressources de u et n à son niveau qualitatif courant. Cette sous-formule signifie que la position de sortie de l'état qualitatif courant est sur le bord supérieur de u . Nous pouvons alors déduire :

$$\begin{aligned} C_{u, \omega, n} \cdot \Delta t &> 1 - \pi_u \\ \Delta t &> \frac{1 - \pi_u}{C_{u, \omega, n}} = \delta_{h'}(u) \quad \text{car } C_{u, \omega, n} > 0 \end{aligned}$$

D'après le lemme 1 page 84, nous avons $\delta_{h'}(v) = \delta_{h'}^{\text{first}} = \Delta t$ et donc $\delta_{h'}(v) > \delta_{h'}(u)$. En d'autres termes, u atteint son bord supérieur avant que v atteigne le sien. Ainsi, la transition continue $h \rightarrow h'$ est telle que $(h, h') \models (\Delta t, a)$ d'après la définition 4.4 page 73.

Les étapes β) et γ) sont identiques au cas $a \equiv \top$.

Troisième cas : $a = \text{noslide}^+(u)$ ou $a = \text{noslide}^-(u)$

α) Similairement au deuxième cas, nous considérons l'unique état hybride $h' = (\eta, \pi')$ tel que la transition continue $h \rightarrow h'$ existe. Le temps passé dans l'état qualitatif courant est aussi Δt (sous-propriété $\Phi_v^+(\Delta t)$). Enfin, de la même manière que dans le deuxième cas, seule une prémisses des conjonctions de la sous-propriété $\mathcal{A}(\Delta t, a)$ est satisfaite. On peut alors remplacer $a = \text{noslide}^+(u)$ par $\text{no}\mathcal{S}^+$:

$$\text{no}\mathcal{S}_{u, \omega, n}^+(\Delta t) \equiv (\pi_u' < 1) \wedge (C_{u, \omega, n} > 0 \Rightarrow \Delta t \leq \frac{\pi_u' - \pi_u}{C_{u, \omega, n}})$$

où ω correspond à l'ensemble des ressources de u et n à son niveau qualitatif courant. Cette sous-formule indique que la position de sortie de l'état qualitatif courant ne peut pas être sur

le bord supérieur de l'entité u . On en déduit :

$$C_{u,\omega,n} > 0 \Rightarrow \Delta t \leq \frac{\pi'_u - \pi_u}{C_{u,\omega,n}} = \delta_h(u)$$

D'après le lemme 1 page 84, $\delta_h^v = \Delta t$ donc $C_{u,\omega,n} > 0 \Rightarrow \delta_h^v \leq \delta_h(u)$: la transition continue $h \rightarrow h'$ est telle que $(h, h') \models (\Delta t, a)$ d'après la définition 4.4 page 73.

Les étapes $\beta)$ et $\gamma)$ sont identiques au cas $a \equiv \top$.

Quatrième cas : $a = C_u \square c$ où $c \in \mathbb{R}$ et \square est l'un des comparateurs habituels

$\alpha)$ La sous-propriété $\mathcal{A}(\Delta t, a)$ permet de remplacer la célérité C_u dans l'assertion a par la célérité indiquée par l'ensemble des ressources et le niveau qualitatif de l'état qualitatif courant. On en déduit $C_{u,\rho(\eta,u),\eta_u} \square c$. A partir de la définition 4.4 page 73, l'unique transition continue $h \rightarrow h'$ où $h' = (\eta, \pi')$ est telle que $(h, h') \models (\Delta t, a)$.

Les étapes $\beta)$ et $\gamma)$ sont identiques au cas $a \equiv \top$.

Cinquième cas : $a = a_1 \wedge a_2$

$\alpha)$ A partir des précédents cas, il est possible de construire deux états hybrides h_1 et h_2 tels que $(h, h_1) \models (\Delta t, a_1)$ et $(h, h_2) \models (\Delta t, a_2)$. Puisque la transition continue démarrante en h est unique, $h_1 = h_2$. Ainsi, $(h, h_1) \models (\Delta t, a_1 \wedge a_2)$.

Cette preuve se généralise pour tous les connecteurs logiques et récursivement à toutes les assertions du langage \mathcal{L}_A .

Les étapes $\beta)$ et $\gamma)$ sont identiques au cas $a \equiv \top$.

La généralisation du cinquième cas aux autres connecteurs logiques permet de faire la preuve de correction pour tout chemin élémentaire, quelque soit l'assertion qu'il contient. Par induction sur la structure d'un chemin, qui se décompose en chemins élémentaires, on prouve donc la correction de la logique de Hoare hybride.

4.5 Complétude sémantique et preuve

Une logique est dite complète si toute formule vraie sémantiquement admet un arbre de preuves construit sur les règles d'inférence : si un triplet de Hoare est satisfait dans un réseau de gènes, alors le triplet de Hoare peut être prouvé en utilisant les règles d'inférence de la logique et en substituant les symboles du triplet par leurs valeurs.

Pour prouver la complétude de la logique de Hoare hybride, nous considérons un triplet de Hoare $\left\{ \begin{array}{l} D_{pre} \\ H_{pre} \end{array} \right\} p \left\{ \begin{array}{l} D_{post} \\ H_{post} \end{array} \right\}$ sémantiquement correct, nous calculons la plus faible précondition

WP à partir de la postcondition et souhaitons prouver que $D_{pre} \wedge H_{pre} \Rightarrow WP$.

Théorème 2. *Pour chaque triplet de Hoare correct $\{Pre\} p \{Post\}$, la plus faible précondition WP donnée par le calcul de plus faible précondition est telle que $Pre \Rightarrow WP$.*

En d'autres mots, la précondition calculée par le calcul proposé est bien la plus faible précondition.

Preuve. Afin de prouver la complétude sémantique de notre logique de Hoare, nous considérons un triplet de Hoare correct, une évaluation des variables η_v , π_v , Δt et $C_{v,\omega,n}$ satisfaisant la précondition Pre , et nous montrons que la formule WP est aussi satisfaite. La preuve est réalisée par induction sur la structure du chemin du triplet de Hoare.

1 - Etape initiale

Nous considérons un chemin élémentaire correspondant à une incrémentation. La preuve pour une décrémentation est similaire.

Considérons le triplet de Hoare correct suivant :

$$\left\{ \begin{array}{l} D_{pre} \\ H_{pre} \end{array} \right\} \left(\begin{array}{c} \Delta t \\ a \\ v+ \end{array} \right) \left\{ \begin{array}{l} D_{post} \\ H_{post} \end{array} \right\}$$

Nous savons que pour tout état $h_1 \models Pre$, il existe un état h_2 tel que $h_1 \xrightarrow{p} h_2$ avec $p = (\Delta t, a, v+)$ et $h_2 \models Post$ (le triplet de Hoare est sémantiquement correct). Le calcul de plus faible précondition mène à :

$$\left\{ \begin{array}{l} D_{wp} \\ H_{wp} \end{array} \right\} \equiv \left\{ \begin{array}{c} D_{post}[\eta_v \setminus \eta_v + 1] \\ H_{post}[\eta_v \setminus \eta_{v+1}] \wedge \Phi_v^+(\Delta t) \wedge \mathcal{F}(\Delta t) \wedge \neg \mathcal{W}_v^+ \wedge \mathcal{A}(\Delta t, a) \wedge \mathcal{J}_v \end{array} \right\}$$

En considérant un état hybride particulier h_1 tel que $h_1 \models Pre$, nous prouvons que $h_1 \models WP$.

$\alpha)$ **Prouvons que $h_1 \models D_{wp}$** Puisque le triplet de Hoare est correct, à partir de l'état hybride h_1 , le chemin élémentaire est possible et mène à un nouvel état hybride h_2 satisfaisant D_{post} . Le chemin élémentaire change uniquement l'état discret η_v de l'atome de chemin discret, permettant de déduire que $h_1 \models D_{wp}$.

$\beta)$ **Prouvons que $h_1 \models H_{post}[\eta_v \setminus \eta_v + 1]$** Puisque la condition hybride après le chemin est satisfaite, et parce que le chemin élémentaire change seulement η_v , nous pouvons déduire que $h_1 \models H_{post}[\eta_v \setminus \eta_v + 1]$.

γ) **Prouvons que** $h_1 \models \Phi_v^+(\Delta t)$ Puisque le chemin élémentaire est possible à partir de h_1 , l'état hybride de sortie de l'état qualitatif courant qui autorise la transition discrète est tel que $\pi_v^{i'} = 1$. De plus, dans l'état qualitatif courant, il existe un unique ensemble de ressources ω tel que Φ_v^ω . Pour tout autre ensemble ω' , les implications sont trivialement vraies. Et pour ω , nous pouvons facilement déduire les deux formules ($C_{v,\omega,n} > 0$) et ($\pi_v^i = \pi_v^{i'} - C_{v,\omega,n} \times \Delta t$), menant à $h_1 \models \Phi_v^+(\Delta t)$.

δ) **Prouvons que** $h_1 \models \mathcal{F}(\Delta t)$ Le chemin élémentaire autorise v à augmenter son niveau d'expression, indiquant alors que v appartient à $\text{first}(h_1)$. Pour les autres entités n'appartenant pas à $\text{first}(h_1)$, trois situations sont possibles :

- ◇ soit la célérité de $u \notin \text{first}(h_1)$ est strictement positive, et si sa célérité autorise u à atteindre son bord avant v , alors u fait face à un mur à son bord supérieur,
- ◇ soit la célérité de $u \notin \text{first}(h_1)$ est strictement négative, et si sa célérité autorise u à atteindre son bord avant v , alors u fait face à un mur à son bord inférieur,
- ◇ soit la célérité de $u \notin \text{first}(h_1)$ est nulle et ne permet pas d'atteindre son bord avant v .

Ainsi, la sous-propriété $\mathcal{F}(\Delta t)$ est satisfaite à l'état hybride h_1 .

ϵ) **Prouvons que** $h_1 \models \neg \mathcal{W}_v^+$ Cette preuve est triviale puisque le chemin élémentaire mène à traverser le bord supérieur de v .

ζ) **Prouvons que** $h_1 \models \mathcal{A}(\Delta t, a)$ Selon l'assertion a , plusieurs cas sont à considérer :

- ◇ si $a \equiv \top$, rien n'est à prouver.
- ◇ si $a \equiv C_u \square c$, nous savons que la célérité de u doit satisfaire la même contrainte, ainsi $\mathcal{A}(\Delta t, a)$ est satisfaite.
- ◇ si $a \equiv \text{slide}^+(u)$, l'assertion est remplacée par $\mathcal{S}_{u,\omega,n}^+(\Delta t) \equiv (\pi_u^{i'} = 1) \wedge (\pi_u^i > \pi_u^{i'} - C_{u,\omega,n} \times \Delta t)$ où ω est l'ensemble des ressources dans l'état qualitatif courant. Cette formule est satisfaite dans l'état hybride courant.
- ◇ si $a \equiv \text{slide}^-(u)$, le raisonnement est similaire au précédent.
- ◇ si $a \equiv \text{slide}(u)$, a est la disjonction des deux précédents cas, donc satisfaite dans l'état courant.
- ◇ si $a \equiv \text{noslide}^+(u)$, l'assertion est remplacée par $\text{no}\mathcal{S}_{u,\omega,n}^+(\Delta t) \equiv (\pi_u^{i'} < 1) \wedge (C_{u,\omega,n} > 0 \Rightarrow \Delta t \leq \frac{\pi_u^{i'} - \pi_u^i}{C_{u,\omega,n}})$ où ω est l'ensemble des ressources dans l'état qualitatif courant. Cette formule est satisfaite dans l'état hybride courant.
- ◇ si $a \equiv \text{noslide}^-(u)$, le raisonnement est similaire au précédent.
- ◇ si $a \equiv \text{noslide}(u)$, a est la conjonction des deux précédents cas, donc satisfaite dans l'état courant.

◇ si $a \equiv \neg\phi$ ou $a \equiv \phi_1 \wedge \phi_2$ ou $a \equiv \phi_1 \vee \phi_2$, la preuve est réalisée par induction structurelle.

η) **Prouvons que** $h_1 \models \mathcal{J}_v$ Pour chaque entité u :

◇ si l'état hybride de sortie de l'état courant autorisant la transition discrète est telle que $\pi_u^{i'} = 1$, la célérité de u doit être positive et la position exacte de l'état hybride d'entrée peut être contraint :

$$\pi_u^i + C_{u,\rho(h,u),\eta_u} \times \delta_h^{first} \geq \pi_u^{i'}$$

indiquant que la célérité de u est telle que la trajectoire atteint le bord supérieur avant la transition discrète.

◇ si l'état hybride de sortie de l'état courant autorisant la transition discrète est telle que $\pi_u^{i'} = 0$, la célérité de u doit être négative et la position exacte de l'état hybride d'entrée peut être contraint :

$$\pi_u^i + C_{u,\rho(h,u),\eta_u} \times \delta_h^{first} \leq \pi_u^{i'}$$

indiquant que la célérité de u est telle que la trajectoire atteint le bord inférieur avant la transition discrète.

◇ dans tous les autres cas, la position exacte de l'état hybride entrant peut être calculée par la formule : $\pi_u^i + C_{u,\rho(h,u),\eta_u} \times \delta_h^{first} = \pi_u^{i'}$

2 - Etape d'induction

Supposons par hypothèse d'induction que l'implication $Pre \Rightarrow WP$ est valide pour tout chemin de longueur inférieure ou égale à n . Considérons un chemin de longueur $n + 1$ composé par un chemin de longueur n appelé p , et concaténé avec un chemin élémentaire. Le triplet de Hoare correct associé à ce chemin est le suivant :

$$\left\{ \begin{array}{l} D_{pre} \\ H_{pre} \end{array} \right\} p; \begin{pmatrix} \Delta t \\ a \\ v+ \end{pmatrix} \left\{ \begin{array}{l} D_{post} \\ H_{post} \end{array} \right\}$$

Considérons un état hybride h_1 tel que $h_1 \models (D_{pre}, H_{pre})$. Puisque le chemin est possible à partir de h_1 , il existe un autre état h_2 tel que $h_1 \xrightarrow{p;(\Delta t, a, v+)} h_2$ et $h_2 \models (D_{post}, H_{post})$. Enfin, considérons l'état h'_2 de ce chemin qui se trouve juste avant le dernier chemin élémentaire. A partir de l'étape initiale, nous déduisons que $h'_2 \models (D'_{post}, H'_{post})$ où (D'_{post}, H'_{post}) est la plus faible précondition calculée à partir de la postcondition (D_{post}, H_{post}) et du dernier chemin élémentaire.

Le triplet $\left\{ \begin{array}{l} D_{pre} \\ H_{pre} \end{array} \right\} p \left\{ \begin{array}{l} D'_{post} \\ H'_{post} \end{array} \right\}$ est correct puisque pour chaque $h_1 \models Pre$, il existe h'_2 tel que $h_1 \xrightarrow{p} h'_2$ et $h'_2 \models (D'_{post}, H'_{post})$. Par hypothèse d'induction, nous avons $h_1 \models (D_{wp}, H_{wp})$ où

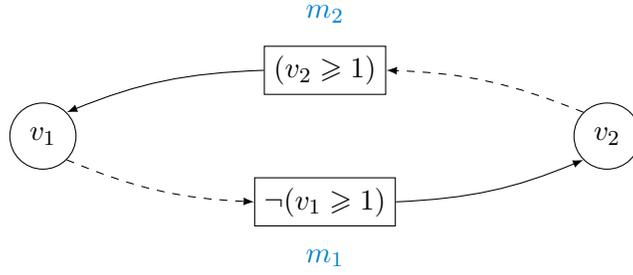


FIGURE 4.7 – Graphe d'influence d'un rétrocontrôle négatif. L'entité v_1 est un inhibiteur de v_2 via le multiplexe m_1 tandis que l'entité v_2 active v_1 via le multiplexe m_2 .

(D_{wp}, H_{wp}) est la plus faible précondition calculée pour la postcondition (D'_{post}, H'_{post}) à travers le chemin p . Mais (D_{wp}, H_{wp}) est aussi la plus faible précondition calculée pour la postcondition (D_{post}, H_{post}) à travers le chemin p ; $\begin{pmatrix} \Delta t \\ a \\ v+ \end{pmatrix}$.

Nous avons donc prouvé que $Pre \Rightarrow WP$. □

Dans cette section, nous avons prouvé la complétude sémantique de la logique de Hoare hybride puisque, pour tout triplet de Hoare correct $\{Pre\} p \{Post\}$, la précondition Pre implique sémantiquement la précondition WP obtenue par le calcul de la plus faible précondition sur la trace biologique p et la postcondition $Post$. Autrement dit, la précondition calculée est bien la plus faible précondition.

4.6 Le cycle négatif développé pas à pas

La logique de Hoare hybride et le calcul de plus faible précondition présentés dans ce chapitre permettent de construire les contraintes sur les paramètres dynamiques du système qui rendent le système compatible avec une trace biologique. Cette approche peut sembler assez technique au premier abord. Pour mieux comprendre la construction de ces contraintes, nous illustrons la démarche sur un exemple classique : le rétrocontrôle négatif qui est très courant en biologie. Il s'agit d'une boucle négative entre plusieurs entités. Dans notre exemple présenté en figure 4.7, nous considérons un rétrocontrôle négatif sur deux entités, v_1 et v_2 . L'entité v_1 inhibe l'entité v_2 via le multiplexe m_1 de formule $\neg(v_1 \geq 1)$ (rappelons qu'un multiplexe agit toujours positivement sur son successeur), tandis que l'entité v_2 active v_1 via le multiplexe m_2 de formule $(v_2 \geq 1)$.

Enfin, nous appliquons la stratégie *backward* sur la trace biologique et la postcondition sui-

vantes :

$$\begin{array}{cccc} \uparrow \begin{pmatrix} 5.0 \\ \text{noslide}(v_1) \\ v_{2+} \end{pmatrix} & ; & \uparrow \begin{pmatrix} 7.0 \\ \text{slide}^+(v_2) \\ v_{1+} \end{pmatrix} & ; & \uparrow \begin{pmatrix} 8.0 \\ \text{noslide}(v_1) \\ v_{2-} \end{pmatrix} & ; & \uparrow \begin{pmatrix} 4.0 \\ \text{slide}^-(v_2) \\ v_{1-} \end{pmatrix} & \left\{ \begin{array}{l} D_4 \\ H_4 \end{array} \right\} \\ (D_0, H_0) & & (D_1, H_1) & & (D_2, H_2) & & (D_3, H_3) & \end{array}$$

avec $D_4 \equiv (\eta_{v_1} = 0) \wedge (\eta_{v_2} = 0)$, $H_4 \equiv \text{True}$. La stratégie *backward* va nous permettre de contraindre successivement les plus faibles préconditions (D_3, H_3) , (D_2, H_2) , (D_1, H_1) et finalement (D_0, H_0) .

Identification du couple (D_3, H_3) Dans un premier temps, la stratégie *backward* identifie la trace biologique comme un chemin non élémentaire p de la forme $p_1; p_2$ avec p_2 le chemin

élémentaire $\begin{pmatrix} 4.0 \\ \text{slide}^-(v_2) \\ v_{1-} \end{pmatrix}$. Remarquons que, à la simple lecture du chemin élémentaire, le temps

passé dans l'état est de 4.0 unités de temps, v_1 décrémente son niveau qualitatif de 1 ($dpa = v_{1-}$), et l'entité v_2 atteint son bord inférieur avant que v_1 atteigne le sien ($a = \text{slide}^-(v_2)$). La stratégie calcule la plus faible précondition sur p_2 (D_3, H_3) qui, d'après la définition 4.7 page 81, peut s'écrire :

$$\begin{aligned} D_3 &\equiv D_4[\eta_{v_1} \setminus \eta_{v_1} - 1], \\ H_3 &\equiv H_4[\eta_{v_1} \setminus \eta_{v_1} - 1] \wedge \Phi_{v_1}^-(\Delta t) \wedge \mathcal{F}(\Delta t) \wedge \neg \mathcal{W}_{v_1}^- \wedge \mathcal{A}(\Delta t, \text{slide}^-(v_2)) \wedge \mathcal{J}_{v_1}; \end{aligned}$$

La condition discrète D_3 est facilement obtenue en remplaçant le niveau qualitatif η_{v_1} par $\eta_{v_1} - 1$ dans D_4 :

$$\begin{aligned} D_3 &\equiv D_4[\eta_{v_1} \setminus \eta_{v_1} - 1] \\ &\equiv (\eta_{v_1} - 1 = 0) \wedge (\eta_{v_2} = 0) \\ &\equiv (\eta_{v_1} = 1) \wedge (\eta_{v_2} = 0) \end{aligned}$$

Puisque les niveaux qualitatifs dans l'état courant sont connus, les célérités des entités peuvent être identifiées. Les célérités agissant dans l'état courant sont $C_{v_1, \emptyset, 1}$ (car $\eta_{v_2} = 0$ donc φ_{m_2} n'est pas satisfaite) et $C_{v_2, \emptyset, 0}$ (car $\eta_{v_1} = 1$ donc φ_{m_1} n'est pas satisfaite). Concernant la condition hybride H_3 , nous appliquons les différentes sous-propriétés présentées en sous-section 4.3.4 page 76 et calculons $H_4[\eta_{v_1} \setminus \eta_{v_1} - 1]$. Le résultat est donné sous forme de tableau pour une meilleure lisibilité, voir tableau 4.1.

Rappelons que, dans les sous-propriétés de la sous-section 4.3.4 page 76, les parties frac-

	H_3 est la conjonction des sous-propriétés
$H_4[\eta_{v_1} \setminus \eta_{v_1} - 1] \equiv$	\top
$\Phi_{v_1}^-(4.0) \equiv$	$(\pi_{v_1}^{3'} = 0) \wedge (C_{v_1, \theta, 1} < 0) \wedge (\pi_{v_1}^3 = \pi_{v_1}^{3'} - C_{v_1, \theta, 1} \times 4.0)$
$\neg \mathcal{W}_{v_1}^- \equiv$	$C_{v_1, \theta, 0} \leq 0$
$\mathcal{A}(4.0, \text{slide}^-(v_2)) \equiv$	$(\pi_{v_2}^{3'} = 0) \wedge (\pi_{v_2}^3 < \pi_{v_2}^{3'} - C_{v_2, \theta, 0} \times 4.0)$
$\mathcal{J}_{v_1} \equiv$	$(\pi_{v_1}^4 = 1 - \pi_{v_1}^{3'}) \wedge (\pi_{v_2}^4 = \pi_{v_2}^{3'}) \wedge$ $(C_{v_1, \theta, 1} < 0) \wedge (\pi_{v_1}^3 \leq \pi_{v_1}^{3'} - C_{v_1, \theta, 1} \times 4.0) \wedge$ $(C_{v_2, \theta, 0} < 0) \wedge (\pi_{v_2}^3 \leq \pi_{v_2}^{3'} - C_{v_2, \theta, 0} \times 4.0)$
$\mathcal{F}(4.0) \equiv$	$C_{v_2, \theta, 0} < 0$

TABLE 4.1 – Calcul de plus faible précondition sur le chemin élémentaire $(4.0; \text{slide}^-(v_2); v_1 -)$. La postcondition correspond au couple (D_4, H_4) .

tionnaires sont décrites dans l'état courant indicé i . Les indices conservés dans cet exemple permettent de différencier les parties fractionnaires entre les différents états.

Les sous-formules du tableau précédent ont été simplifiées en utilisant le fait que dans chaque sous-formule, une unique prémisse des implications est satisfaite. En effet, la prémisse peut être correcte ou non selon l'état courant. Ici, l'état courant est caractérisé par : $\eta_{v_1} = 1$, $\eta_{v_2} = 0$, ce qui implique qu'il n'y a aucune ressource, ni pour v_1 , ni pour v_2 .

La première formule consiste à remplacer η_{v_1} par $\eta_{v_1} - 1$ dans la condition hybride H_4 . Puisque $H_4 \equiv \top$, on en déduit que $H_4[\eta_{v_1} \setminus \eta_{v_1} - 1] \equiv \top$.

Ensuite, la sous-propriété $\Phi_{v_1}^-(4.0)$ décrit les conditions pour lesquelles l'entité v_1 atteint son bord inférieur pour le franchir. Ainsi, la position de sortie de v_1 est $\pi_{v_1}^{3'} = 0$, sa célérité est négative ($C_{v_1, \theta, 1} < 0$) et le délai de contact avant d'atteindre un bord de v_1 est de 4.0 unités de temps ($\pi_{v_1}^3 = \pi_{v_1}^{3'} - C_{v_1, \theta, 1} \times 4.0$).

La sous-propriété suivante est $\neg \mathcal{W}_{v_1}^-$. Elle empêche v_1 de faire face à un mur, assurant à v_1 de pouvoir traverser son bord. En simplifiant les formules ($\neg \text{EW}_{v_1}^- \equiv \top$), on déduit que $\neg \mathcal{W}_{v_1}^- \equiv \neg \text{IW}_{v_1}^- \equiv (C_{v_1, \theta, 1} \geq 0) \vee (C_{v_1, \theta, 0} \leq 0)$. Or la sous-propriété $\Phi_{v_1}^-(4.0)$ a montré qu'on avait $C_{v_1, \theta, 1} < 0$ dans l'état courant. On en conclut $\neg \mathcal{W}_{v_1}^- \equiv C_{v_1, \theta, 0} \leq 0$.

La sous-propriété $\mathcal{A}(4.0, \text{slide}^-(v_2))$, traduisant les symboles d'assertion, assure à l'entité v_2 d'atteindre son bord avant l'entité du *dpa* v_1 . Cette sous-propriété revient à calculer la sous-formule $\mathcal{S}_{v_2, \theta, 0}^-(4.0)$. Ainsi, $\mathcal{A}(4.0, \text{slide}^-(v_2)) \equiv (\pi_{v_2}^{3'} = 0) \wedge (\pi_{v_2}^3 < \pi_{v_2}^{3'} - C_{v_2, \theta, 0} \times 4.0)$. Remarquons que la célérité $C_{v_2, \theta, 0}$ est strictement négative puisque $\pi_{v_2}^{3'} = 0$ et $C_{v_2, \theta, 0} \times 4.0 < \pi_{v_2}^{3'} - \pi_{v_2}^3$.

La sous-propriété \mathcal{J}_{v_1} fait le lien entre l'état courant et l'état de postcondition (jonction discrète), et évalue l'évolution de chaque entité durant la transition continue (jonctions continues). Ainsi $\mathcal{J}_{v_1} \equiv \mathcal{D}\mathcal{J}_{v_1} \wedge \bigwedge_{u \in V} \mathcal{C}\mathcal{J}_u$, avec $\mathcal{D}\mathcal{J}_{v_1} \equiv (\pi_{v_1}^4 = 1 - \pi_{v_1}^{3'}) \wedge (\pi_{v_2}^4 = \pi_{v_2}^{3'})$ et

$\mathcal{CJ}_v \equiv (C_{v_1, \emptyset, 1} < 0) \wedge (\pi_{v_1}^3 \leq \pi_{v_1}^{3'} - C_{v_1, \emptyset, 1} \times 4.0) \wedge (C_{v_2, \emptyset, 0} < 0) \wedge (\pi_{v_2}^3 \leq \pi_{v_2}^{3'} - C_{v_2, \emptyset, 0} \times 4.0)$ puisque les positions de sortie des entités sont connues.

Finalement, la sous-propriété $\mathcal{F}(4.0)$ contraint les entités qui ne sont pas dans **first** à atteindre leur bord après les entités du **first**, ou bien à faire face à un mur. Nous savons que $C_{v_2, \emptyset, 0} < 0$ par la sous-propriété $\mathcal{A}(4.0, \text{slide}^-(v_2))$. Cette information revient à dire que $\mathcal{F}(4.0) \equiv \mathcal{W}_{v_1}^-$. Puisque $\eta_{v_2} = 0$, on en déduit que $\mathcal{F}(4.0) \equiv C_{v_2, \emptyset, 0} < 0$.

En simplifiant les contraintes obtenues entre les différentes sous-proprietés, on en déduit :

$$\begin{aligned} H_3 \equiv & (C_{v_1, \emptyset, 1} < 0) \wedge (C_{v_2, \emptyset, 0} < 0) \wedge (C_{v_1, \emptyset, 0} \leq 0) \wedge \\ & (\pi_{v_1}^{3'} = 0) \wedge (\pi_{v_2}^{3'} = 0) \wedge (\pi_{v_1}^4 = 1 - \pi_{v_1}^{3'}) \wedge \\ & (\pi_{v_2}^4 = \pi_{v_2}^{3'}) \wedge (\pi_{v_1}^3 = \pi_{v_1}^{3'} - C_{v_1, \emptyset, 1} \times 4.0) \wedge \\ & (\pi_{v_2}^3 < \pi_{v_2}^{3'} - C_{v_2, \emptyset, 0} \times 4.0) \end{aligned}$$

Nous avons donc identifié le couple (D_3, H_3) , la plus faible précondition pour le chemin élémentaire $\begin{pmatrix} 4.0 \\ \text{slide}^-(v_2) \\ v_1- \end{pmatrix}$ et la postcondition présentée plus haut. Le calcul de plus faible précondition est alors réitéré pour identifier le couple (D_2, H_2) . La postcondition utilisée pour ce calcul correspond au couple (D_3, H_3) et le chemin élémentaire est $(8.0, \text{slide}^-(v_2), v_1-)$. Ce procédé est répété successivement sur les trois chemins élémentaires restants jusqu'à obtenir le couple (D_0, H_0) .

Identification du couple (D_0, H_0) En procédant selon la même méthode décrite précédemment, nous avons pu identifier la plus faible précondition de la trace biologique étudiée. La condition discrète D_0 obtenue est $(\eta_{v_1} = 0) \wedge (\eta_{v_2} = 0)$. La condition hybride H_0 présente 38 contraintes dont 6 égalités, voir annexe 9.1 page 149. Ce système de contraintes comprend 8 célérités et 18 parties fractionnaires. En effet, 4 états qualitatifs sont parcourus durant ce chemin, impliquant 8 célérités à identifier. Les couples non finaux $(D_i, H_i)_{i \in [0..3]}$ présente 4 parties fractionnaires : 2 parties fractionnaires pour l'état hybride d'entrée et 2 pour l'état hybride de sortie. De plus, le couple final (D_4, H_4) présente 2 parties fractionnaires pour l'état hybride d'entrée. Les simplifications effectuées à la volée permettent de déterminer 6 parties fractionnaires de sortie. En effet, puisque 4 états qualitatifs sont parcourus dans cette trace biologique, la sous-propriété $\Phi_v^\pm(\Delta t)$ permet de connaître la position de sortie de l'entité v du dpa (0 ou 1, v étant sur un bord). Ainsi 4 parties fractionnaires de sortie ($\pi_v' = 1$ pour $\Phi_v^+(\Delta t)$, $\pi_v' = 0$ pour $\Phi_v^-(\Delta t)$) sont trivialement identifiables. Les deux dernières parties fractionnaires de sorties identifiées correspondent aux glissements de l'entité v_2 , signifiant que cette entité atteint son

bord avant le prochain événement ($\pi'_{v_2} = 1$ pour $\text{slide}^+(v_2)$, $\pi'_{v_2} = 0$ pour $\text{slide}^-(v_2)$).

Notons aussi que les contraintes obtenues ne sont pas linéaires dans le cas général car le temps de chacun des chemins élémentaires Δt est multiplié avec une célérité et que ce temps et cette célérité peuvent être tous deux inconnues. Cette situation arrive lorsque l'on sait que les deux premiers chemins élémentaires sont traversés en 12.0 unités de temps, sans connaître le temps exact nécessaire à chacun d'eux. Nous indiquons alors dans la condition hybride H_4 de la postcondition la contrainte supplémentaire $\Delta t_1 + \Delta t_2 = 12.0$ où Δt_1 (resp. Δt_2) correspond au temps nécessaire pour parcourir le premier (resp. second) chemin élémentaire.

En résumé, un automate hybride peut modéliser un système biologique et ses traces peuvent être des représentations des traces biologiques expérimentales. Les paramètres dynamiques nommés célérités pilotent la dynamique de ces automates hybrides et doivent être identifiés pour rendre le modèle cohérent avec les traces observées. La logique de Hoare initialement développée pour la preuve de programmes impératifs a été modifiée dans l'objectif de vérifier la cohérence du modèle avec les traces observées. La logique de Hoare hybride munie du calcul de la plus faible précondition permet ainsi de construire les contraintes nécessaires sur les paramètres dynamiques pour que le modèle soit en accord avec la trace biologique observée. Ce système formel a été prouvé correct et complet.

L'étape d'après consiste à résoudre les contraintes obtenues, c'est-à-dire à identifier un ou plusieurs jeux de paramètres satisfaisant les contraintes : c'est un problème de satisfaction de contraintes encore appelé **CSP** (Constraint Satisfaction Problem). Le comportement des entités du graphe pourra ainsi être simulé (en utilisant les valeurs des paramètres précédemment trouvées) et comparé aux comportements expérimentaux.

Le chapitre suivant décrit comment utiliser des solveurs de contraintes pour trouver des solutions au CSP issues du calcul de plus faible précondition.

Chapitre 5

Résolution des contraintes grâce à un solveur

Le chapitre précédent a permis de construire les contraintes minimales sur les paramètres d'un modèle pour qu'il soit cohérent avec une trace observée. Néanmoins, la construction de ces contraintes ne suffit pas à trouver les valeurs des paramètres compatibles. La résolution "à la main" de ces contraintes, envisageable pour des graphes d'influence de petite taille, se révèle impossible lorsque le nombre d'entités du graphe augmente. Dans ce chapitre, nous montrons comment la résolution de ces contraintes peut être automatisée en utilisant des solveurs de contraintes. Nous illustrons notre démarche sur un exemple.

5.1 Introduction sur les méthodes de solveurs

Les **problèmes de satisfaction de contraintes (CSP)** sont des problèmes couramment observés dans notre vie quotidienne : planification du trafic aérien dans un aéroport [86], mise en place d'un planning dans les écoles [134], ou encore la prédiction de structure secondaire d'ARN [104]. Chacun de ces exemples met en jeu un ensemble de contraintes qu'on cherche à résoudre. La recherche de solutions varie selon les problèmes posés : parfois il s'agit de trouver une unique solution ou l'ensemble des solutions satisfaisant les contraintes, dans d'autres cas, il s'agit de trouver une solution optimale en minimisant un critère (par exemple, minimiser la somme des distances entre les correspondances pour un vol aérien), ou encore prouver qu'aucune solution n'existe.

Les CSP sont généralement représentés par des triplets de la forme (V, D, C) où V correspond à l'ensemble des variables du système de contraintes, D aux domaines de définition de ces variables et C à l'ensemble des contraintes exprimées par des formules logiques du premier ordre. Les

domaines des variables sont dits *discrets* lorsque les variables sont discrètes, et *continus* pour les variables réelles. Les contraintes sont des formules logiques dont les formules atomiques sont des égalités ou inégalités dont les expressions droite et gauche sont des expressions arithmétiques construites sur les constantes de \mathbb{N} ou \mathbb{R} et sur les variables du CSP en utilisant les opérateurs arithmétiques habituels. Le but du jeu consiste alors à trouver comment attribuer une valeur ou un ensemble de valeurs à chaque variable du système, selon leurs domaines de définition, tout en conservant la satisfaction des contraintes : on parle d'**affectation**. Une solution du CSP est trouvée lorsque chaque variable du système a une affectation.

Assez naturellement, puisque la résolution du CSP concerne l'ensemble des contraintes du système, les contraintes sont souvent représentées sous *forme normale conjonctive*, c'est-à-dire que l'expression logique est sous la forme d'une conjonction de clauses disjonctives (exemple : $(A = 2.0) \wedge (A > 1.0 \vee B \geq 0.0) \wedge (C \neq 3.0)$).

Les CSP, munis de diverses méthodes de résolution de contraintes, forment le paradigme de programmation appelé **programmation par contraintes**. Dans ce paradigme, les aspects fonctionnels (CSP) et techniques (algorithme de résolution) sont séparés :

"Constraint Programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming : the user states the problem, the computer solves it."

Eugene C. Freuder, Inaugural issue of the Constraints Journal, 1997

Cette citation célèbre d'Eugene Freuder démontre à quel point la programmation par contraintes a étendu la programmation logique [34]. En séparant les aspects fonctionnels des aspects techniques, on s'éloigne de la programmation classique pour mettre en avant une nouvelle approche de programmation.

Les logiciels qui résolvent les CSPs sont appelés **solveurs de contraintes**. L'ensemble des contraintes est spécifié au solveur qui se charge de résoudre ce CSP ou prouver qu'il n'existe pas de solution. Néanmoins, les algorithmes classiques ne permettent pas de parcourir de manière exhaustive l'ensemble des combinaisons possibles des valeurs des variables. Des raisonnements et heuristiques sont intégrés aux solveurs : ce sont les concepts fondamentaux de **propagation de contraintes** (encore appelé filtrage) et d'**exploration**. Les valeurs menant à une incohérence dans le CSP sont filtrées et éliminées par des déductions logiques, tandis que des heuristiques explorent l'espace de recherche à partir d'hypothèses réfutables. En effet, des hypothèses en contradiction avec les contraintes sont formulées, les valeurs satisfaisant ces hypothèses ne pourront pas être solution du CSP, ce qui assure de pouvoir éliminer ces valeurs dans les domaines de définition des variables correspondantes, sans changer l'ensemble des solutions.

La propagation de contraintes consiste à réduire le domaine des variables d'un CSP de telle

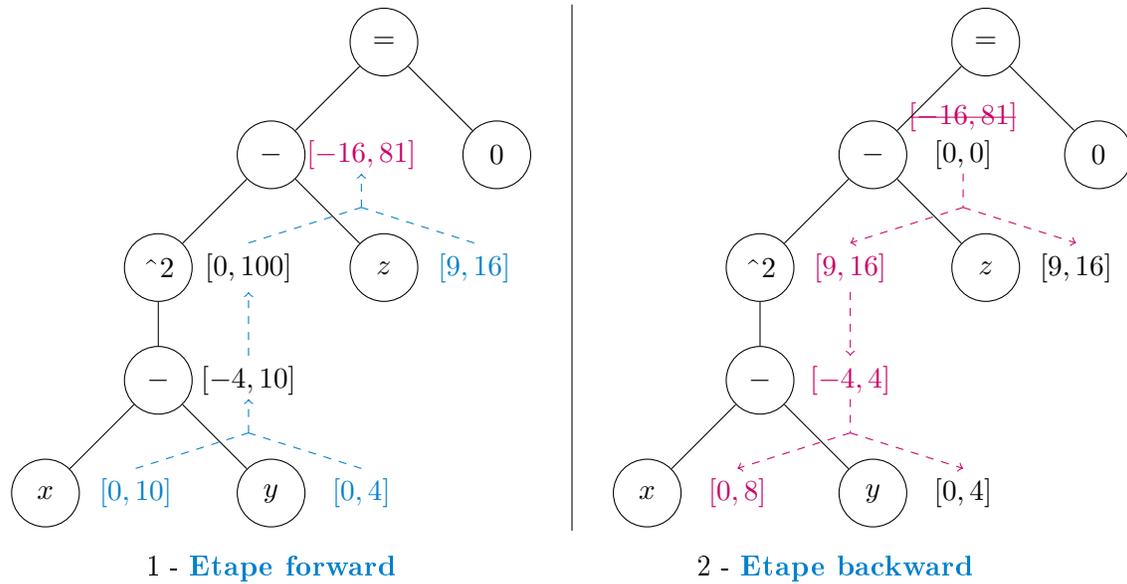


FIGURE 5.1 – Illustration de l’algorithme HC4-Revise sur l’équation $(x - y)^2 - z = 0$. L’arbre de gauche décrit l’étape forward pour laquelle les intervalles des variables sont propagés vers la racine. L’étape backward (arbre de droite) réduit les intervalles en propageant le résultat attendu de la racine vers les feuilles de l’arbre.

sorte que les valeurs supprimées ne peuvent pas prendre part à une solution du CSP initial. De nombreux algorithmes de réduction des domaines de définition existent. En continu, l’algorithme le plus connu est HC4-Revise [16] qui utilise l’arithmétique des intervalles [94]. Pour comprendre l’intuition de cet algorithme, rappelons que la contrainte est représentée par un arbre syntaxique dont les feuilles sont les variables du CSP et les nœuds internes des opérateurs arithmétiques et dont la racine est un comparateur, voir figure 5.1. Les intervalles de chaque variable sont propagés des feuilles en direction de la racine et sont modifiés selon les opérations attachées aux nœuds (étape forward). L’intervalle obtenu par cette étape pour le membre de gauche de l’(in)égalité est comparé avec celui obtenu pour le membre droit (l’intervalle réduit à la valeur 0 dans le cas de la figure 5.1), puis cette information est propagée aux intervalles des différents nœuds ce qui peut permettre de les réduire (étape backward).

Exemple: Dans l’exemple de la figure 5.1¹, la contrainte considérée est $(x - y)^2 - z = 0$ est présentée sous forme d’arbre, et les domaines de définition des variables sont $D_x = [0, 10]$, $D_y = [0, 4]$ et $D_z = [9, 16]$. L’étape forward (figure de gauche) propage les intervalles vers la racine. L’intervalle de définition pour l’expression $x - y$ devient $[-4, 10]$. La propagation est répétée jusqu’à la racine et le domaine de définition de l’expression $(x - y)^2 - z$ devient $[-16, 81]$.

1. Les figures 5.1 et 5.2 page 101 sont des exemples provenant du site <http://www.ibex-lib.org/doc/contractor.html>

Or l'égalité impose à l'expression $(x - y)^2 - z$ de valoir 0. L'intervalle $[-16, 81]$ est donc réduit dans l'étape backward (figure de droite). L'information est propagée et le domaine de définition de l'expression $(x - y)^2$ devient $[9, 16]$. En réitérant le procédé, le domaine de définition de la variable x est réduit de $[0, 10]$ à $[0, 8]$.

Cet exemple de propagation de contraintes démontre qu'il est possible de réduire les domaines de définition des variables pour identifier l'ensemble des solutions du CSP. Cependant, lorsque le nombre d'occurrences de variables dans l'expression augmente, la contraction des intervalles perd en précision. Cet algorithme s'intègre alors dans d'autres stratégies de propagation comme la boucle de *Propagation* [20]. L'idée générale de cet algorithme consiste à identifier les plus petits intervalles, associés aux variables, dont le produit cartésien contient toutes les solutions du CSP. Pour raisonner sur les intervalles des variables, on note $[x]$ un quelconque intervalle du domaine de définition de x . Les fonctions qui prennent en argument un intervalle $[x]$ et qui le réduisent, sont appelées **propagateurs**. Et la boucle de *Propagation* calcule le **point fixe** des propagateurs à partir d'un produit cartésien d'intervalles initiaux. Les propagateurs s'interprètent comme des algorithmes qui réduisent le domaine de définition d'une variable x en un sous-domaine qui contient toutes les solutions de la contrainte courante, voir figure 5.2 page ci-contre. Un propagateur P doit vérifier les quatre propriétés suivantes :

- ◇ la *correction* : $[x] \cap Sc \subseteq P([x])$ où Sc est l'ensemble des solutions de la contrainte c ,
- ◇ la *contraction* : $\forall [x] \in \mathbb{R}^n, P([x]) \subseteq [x]$,
- ◇ la *monotonie* : $\forall [x] \subseteq [x'], P([x]) \subseteq P([x'])$,
- ◇ l'*idempotence* : $P(P([x])) = P([x])$

où $P([x])$ désigne l'application du propagateur P sur $[x]$.

En d'autres termes, le propagateur P identifie un domaine de définition de x contenant l'ensemble des solutions de la contrainte donnée dans l'intervalle initial $[x]$ (*correction*), peut réduire ce domaine de définition (*contraction*), doit être monotone et l'appel du propagateur P sur $P([x])$ n'a aucun effet (*idempotence*).

Au final, pour un CSP (V, D, C) , la *Propagation* réduit au maximum tous les domaines de définition D des variables dans V , et peut indiquer qu'il n'existe pas de solution, le cas échéant.

Les CSPs définis à partir de la logique de Hoare hybride contiennent des variables réelles. En effet, les variables des CSPs construits sont les célérités (paramètres dynamiques) définies sur \mathbb{R} et les parties fractionnaires définies sur $[0, 1]$. Les solutions des solveurs continus sont des produits d'intervalles appelées **boîtes**. Celles ne contenant que des solutions du CSP sont appelées **boîtes sûres**, tandis que celles contenant au moins une solution du CSP sont appelées **boîtes non sûres**, voir figure 5.3 page suivante. Notons que la taille de ces boîtes dépend du paramètre de précision donné au solveur.

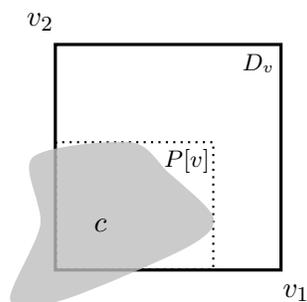


FIGURE 5.2 – Schéma simplifié d'un sous-domaine issu du propagateur P . Les domaines de définition des variables v_1 (D_{v_1}) et v_2 (D_{v_2}) sont réduits par l'algorithme P et conserve les valeurs de v_1 et v_2 satisfaisant la contrainte c .

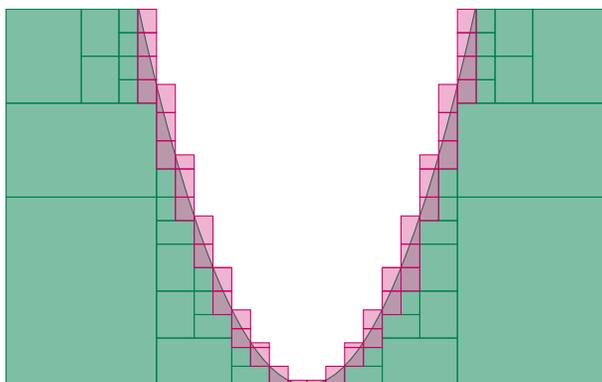


FIGURE 5.3 – Comparaison entre les solutions sûres et non sûres de la contrainte $y \leq x^2$. Les boîtes vertes contiennent uniquement des solutions (boîtes sûres) tandis que les boîtes roses contiennent au moins une solution (boîtes non sûres).

Exemple de boîte non sûre: Prenons l'exemple de la contrainte $x - x = 0$ avec $D_x = [-1, 1]$. Puisque les algorithmes de résolution propagent les intervalles dans les contraintes du CSP, les occurrences des mêmes variables dans une contrainte sont décorréées. Ainsi, chaque occurrence de la variable dans la contrainte précédente est interprétée différemment, et la contrainte est réécrite sous la forme $x_1 - x_2 = 0$ où les domaines de définition de ces 2 variables sont égaux et valent $[-1, 1]$. L'égalité est satisfaite lorsque $x_1 = x_2$ et la stratégie forward-backward mène à donner comme domaines solutions : $D_{x_1} = D_{x_2} = [-1, 1]$. Pourtant, affecter la valeur -1 à x_1 et la valeur 1 à x_2 ne satisfait pas la contrainte. Les solveurs sont alors incertains de la solution fournie et retournent une boîte non sûre.

Dans la suite de ce chapitre, nous présentons les solveurs continus Ibex [98] et AbSolute [103] que nous avons utilisé afin de résoudre les CSPs construits par la stratégie backward de la logique de Hoare hybride à partir de traces biologiques. Ces CSPs (V, D, C) sont définis par les

3 éléments suivants :

- ◇ les variables du CSP sont les célérités, les parties fractionnaires d'entrée et de sortie dans chacun des états qualitatifs de la trace observée et éventuellement le temps passé dans certains états qualitatifs,
- ◇ les célérités sont définies sur \mathbb{R} , les parties fractionnaires sur $[0, 1]$ et le temps sur \mathbb{R}^{+*} ,
- ◇ les contraintes sont celles obtenues en appliquant la stratégie backward sur le calcul de plus faible précondition décrite en section 4.3 page 69.

Enfin, nous décrirons un exemple de résolution de contraintes sur le cycle négatif associé à une trace biologique et une postcondition, toutes deux présentées en section 4.6 page 92.

5.2 Premier essai avec Ibex

Pour résoudre les contraintes du CSP, notre premier choix s'est porté sur le solveur continu **Ibex** [98]. En plus de résoudre des contraintes linéaires sur des nombres réels, Ibex résout également des contraintes non linéaires qu'il est possible de retrouver dans notre CSP (exemple : $\pi_v = \pi'_v - C_{v,0,0} \times \Delta t$). Nous avons choisi ce solveur pour ses nombreux propagateurs qu'il est possible de combiner spécifiquement afin de résoudre notre système de contraintes, et parce que la résolution de contraintes non linéaires est intéressante lorsque le temps passé dans un chemin élémentaire n'est pas une donnée expérimentale.

Le solveur par défaut d'Ibex est nommé *IbexSolve*. Ce solveur se compose d'un propagateur pour la propagation des contraintes et d'un bissecteur pour l'exploration. Un **bissecteur** est un opérateur prenant une boîte $[v]$ en entrée et qui retourne deux boîtes $[v_1]$ et $[v_2]$ telles que $[v] = [v_1] \cup [v_2]$.

La propagation utilisée dans le solveur par défaut regroupe différents algorithmes :

- ◇ HC4-revise qui propage les contraintes [16],
- ◇ ACID (Adaptative Constructive Interval Disjunction) qui calcule dynamiquement la prochaine variable qui sera coupée en intervalles plus petits selon l'impact qu'à cette variable sur les contraintes [129, 97]. L'impact d'une variable v sur une fonction f est la dérivée partielle de f par rapport à v , multipliée par la taille de l'intervalle $[v]$. La variable choisie est celle où l'impact est le plus grand.
- ◇ l'algorithme Polytope Hull qui calcule le plus petit polytope contenant l'ensemble des points satisfaisant les contraintes.

Enfin, le bissecteur se base sur la fonction *Smear* qui identifie la prochaine variable à couper selon l'impact qu'a l'ensemble des contraintes sur cette variable [128].

La description des contraintes du CSP pour le solveur Ibex est donnée par un fichier qui comporte deux parties. La première partie nommée *Variables* décrit l'ensemble des variables

V avec leurs domaines de définition correspondants. La seconde partie, nommée *Constraints*, décrit le système de contraintes C où toutes les contraintes sont en conjonction.

Puisque Ibex ne prend pas en compte les clauses disjonctives, nous avons écrit les contraintes sous forme normale disjonctive (FND) puis nous avons créé un fichier par clause conjonctive.

Les premiers résultats que nous avons obtenus ont été infructueux en raison de l'explosion combinatoire des disjonctions présentes dans les contraintes. Par exemple, en reprenant le graphe d'influence comprenant 2 entités v_1 et v_2 présenté en figure 4.7 page 92 et la trace biologique

$$\left(\begin{array}{c} 5.0 \\ \top \\ v_{2+} \end{array} \right); \left(\begin{array}{c} 7.0 \\ \top \\ v_{1+} \end{array} \right); \left(\begin{array}{c} 8.0 \\ \top \\ v_{2-} \end{array} \right); \left(\begin{array}{c} 4.0 \\ \top \\ v_{1-} \end{array} \right) \left\{ \begin{array}{l} D_4 \\ H_4 \end{array} \right\}$$

avec $D_4 \equiv (\eta_{v_1} = 0) \wedge (\eta_{v_2} = 0)$, $H_4 \equiv True$, le calcul de la plus faible précondition mène à 8 disjonctions, ce qui se traduit par la création de 256 fichiers. Ces disjonctions proviennent de la sous-propriété $\mathcal{F}(\Delta t)$ lorsqu'une entité n'est jamais contrainte que ce soit par les symboles d'assertion ou par les autres chemins élémentaires de la trace biologique. Pour cet exemple, la quantité de fichiers à analyser par le solveur reste abordable. Toutefois, pour le graphe d'influence du cycle circadien comprenant 4 entités et dont une trace biologique est constituée de 8 chemins élémentaires (voir section 6.5.1 page 123), le calcul de plus faible précondition contient 42 disjonctions, ce qui mènerait à la construction d'environ 4400 milliards fichiers (2^{42}) !

Ces premiers essais montrent que la transformation des contraintes obtenues par le calcul de plus faible précondition en FND mène à trop de fichiers pour que le solveur Ibex puisse être efficace. Nous nous sommes tournés vers un nouveau solveur continu plus adapté aux CSPs construits par le calcul de plus faible précondition de la logique de Hoare hybride : le solveur *AbSolute*.

5.3 Les avancées grâce au solveur AbSolute

AbSolute est un solveur de contraintes linéaires et non linéaires sur les réels qui inclut les clauses disjonctives [103]. Ce solveur s'appuie sur la notion de domaines abstraits qui sur-approxime l'ensemble des solutions satisfaisant les contraintes. Le solveur AbSolute propage également les domaines de définition des variables sur les contraintes et réduit la taille de ces intervalles par contraction. AbSolute propose de fournir en entrée une précision (la taille minimale des intervalles des variables) ainsi que le nombre de solutions à identifier.

5.3.1 Réécriture symbolique

Avant la propagation des domaines de définition, une étape de preprocessing est nécessaire pour simplifier les contraintes d'égalité. La taille du problème peut être réduite en éliminant des variables par **réécriture symbolique**.

La première étape de ce preprocessing consiste à identifier l'ensemble des constantes du CSP, c'est-à-dire l'ensemble des variables v pour lesquelles il existe une contrainte de la forme $v = n$ avec $n \in \mathbb{R}$, et de les conserver dans une liste. On remplace dans tout le CSP les constantes par leurs valeurs et on réitère le processus jusqu'à atteindre un point fixe.

La seconde étape du preprocessing consiste à identifier l'ensemble des contraintes d'égalité entre deux variables (de la forme $\pi_{v_1}^4 = 1 - \pi_{v_1}^{3'}$) et de les conserver dans une liste de **vues de variables**. Une variable est choisie au hasard dans une des vues et toutes ses occurrences dans le CSP sont remplacées par sa réécriture issue de cette contrainte et des contraintes sur les bornes sont ajoutées (dans l'exemple précédent, on remplace $\pi_{v_1}^4$ par $1 - \pi_{v_1}^{3'}$). Le processus est répété jusqu'à atteindre un point fixe.

Illustrons ce preprocessing sur le CSP (V, D, C) suivant :

- ◊ $V = \{v_1, v_2, v_3\}$,
- ◊ $D = \{D_{v_1} = [-5, 5], D_{v_2} = [-5, 5], D_{v_3} = [-5, 5]\}$,
- ◊ $C = \{c_1 : v_1 = 2, \quad c_2 : v_1 = v_2 + v_3, \quad c_3 : v_3 = 3 \times (v_2)^2\}$,
- ◊ $vues = \{\}$.

D'après la contrainte c_1 , v_1 est une constante. La première étape remplace v_1 par la valeur 2 dans c_2 qui devient une vue de variable : v_2 est alors remplacée par $2 - v_3$ dans c_3 et on contraint $2 - v_3$ par les bornes de l'intervalle de v_2 . Ainsi on obtient un nouveau CSP :

- ◊ $V' = \{v_3\}$,
- ◊ $D' = \{D_{v_3} = [-5, 5]\}$,
- ◊ $C' = \{c'_1 : v_3 = 3 \times (2 - v_3)^2, \quad c'_2 : 2 - v_3 \geq -5, \quad c'_3 : 2 - v_3 \leq 5\}$,
- ◊ $vues = \{v_2 = 2 - v_3\}$.

Le CSP a été réduit de trois variables à une seule, une constante $v_1 = 2$ et une vue de variable $v_2 = 2 - v_3$.

5.3.2 Domaines de définition invariants et changeants

Le solveur AbSolute a été modifié pour s'adapter aux CSPs issus de la logique de Hoare en temps continu. Chaque solution est représentée par deux ensembles de domaines de définition :

- ◊ le premier ensemble contient les domaines des variables sûres, c'est-à-dire que, quelle que soit les valeurs prises de ces variables dans leurs domaines de définition, elles satisfont toutes les contraintes du CSP. Par abus de langage, dû au choix d'implémentation, on

parle de **liste d'invariants** LI .

- ◊ le second ensemble contient les domaines de définition des variables dont au moins une valeur peut satisfaire les contraintes du CSP. On parle de **liste de changeants** LC .

Intuitivement, l'ensemble des domaines de définition d'un CSP (V, D, C) correspond à l'union de la liste d'invariants et de la liste de changeants ($D = LI \cup LC$). Par abus de langage, nous appelons **variable invariante** (resp. variable changeante) les variables dont le domaine de définition est dans LI (resp. LC).

Lorsqu'une solution est sûre, sa liste de changeants est vide. A l'inverse, une solution non sûre présente au moins un domaine de variable dans la liste de changeants.

Lorsqu'une solution du CSP est identifiée, une des deux variables de chaque vue notée x est remplacée par son expression dépendant de l'autre variable notée y . Intuitivement, si y est une variable invariante (resp. variable changeante), x est aussi une variable invariante (resp. variable changeante).

Exemples de LI et LC sur un CSP:

Soit (V, D, C) un CSP tel que :

- ◊ $V = \{v_1, v_2, v_3, v_4\}$
- ◊ $D = LI \cup LC$
 - où $LI = \{\}$ et $LC = \{D_1 = [-5, 5], D_2 = [-5, 5], D_3 = [-5, 5], D_4 = [-5, 5]\}$
- ◊ $C = \{c_1 : v_1 = 2, c_2 : v_1 = v_2 + v_3, c_3 : v_3 = 3 \times (v_2)^2, c_4 : v_4 < -1\}$
- ◊ $vues = \{\}$

Initialement, tous les domaines de définition des variables appartiennent à LC , donc $LI = \emptyset$. La réécriture symbolique de ces contraintes est identique à celle présentée en sous-section 5.3.1 page ci-contre. La contrainte $v_1 = v_2 + v_3$ devient $v_2 = 2 - v_3$ qui est une vue de variable. Les contraintes du nouveau CSP deviennent alors $C' = \{c'_1 : v_3 = 3 \times (2 - v_3)^2, c'_2 : 2 - v_3 \geq -5, c'_3 : 2 - v_3 \leq 5\}$. Notons que la contrainte $c_4 : v_4 < -1$ n'est plus présente car c'est la seule contrainte sur v_4 , menant à définir v_4 comme variable invariante sur le domaine $[-5, -1[$. Puisque les variables v_1 et v_4 ne sont plus définies dans ces contraintes, leurs domaines de définition sont retrouvés dans la liste d'invariants et sont réduits selon les contraintes des variables (c_1 et c_4).

La variable v_2 participe à la vue : $v_2 = 2 - v_3$, v_2 est remplacée dans la contrainte $v_3 = 3 \times (v_2)^2$ par $(2 - v_3)$. Puisque v_3 est une variable changeante (aucune information sur v_3 au départ), le domaine de définition de la variable v_2 est aussi dans LC . Enfin, l'expression $2 - v_3$ est contrainte par les bornes de l'intervalle de v_2 . Le CSP obtenu final est donc :

- ◊ $V' = \{v_1, v_2, v_3, v_4\}$
- ◊ $D' = LI \cup LC$
 - où $LI = \{D_{v_1} = [2, 2], D_{v_4} = [-5, -1]\}$ et $LC = \{D_{v_2} = [-3, 5], D_{v_3} = [-5, 5]\}$

- ◇ $C' = \{c_1 : v_3 = 3 \times (2 - v_3)^2, c_2 : 2 - v_3 \geq -5, c_3 : 2 - v_3 \leq 5\}$
- ◇ $vues = \{v_2 = 2 - v_3\}$

5.3.3 Limites du solveur

Comme tout solveur de contraintes, AbSolute comporte des limitations qui peuvent ralentir voire empêcher de résoudre le CSP initial. Dans le cas des CSPs issus du calcul de plus faible précondition de la logique de Hoare hybride, certaines de ces limitations peuvent être énumérées :

- ◇ Lorsque le solveur de contraintes permet de trouver facilement des solutions du problème posé, le temps de calcul peut être ralenti en voulant récupérer un trop grand nombre de solutions.
- ◇ L'approximation des réels en flottants crée de nombreux problèmes dans la résolution des contraintes. Des domaines de définition dans la liste de changeants pourraient donner des solutions qui ne satisfont pas les contraintes (exemple de variable v dont la valeur est fixée dans son domaine de définition à 2.500001 alors que cette variable doit satisfaire la contrainte $v = 5.0/2.0$). La réécriture symbolique a permis de limiter ce problème car elle peut simplifier les contraintes en factorisant les variables identiques.
- ◇ Lorsqu'on améliore la précision sur les intervalles des variables du CSP, le temps de calcul augmente. En effet, tant qu'une solution n'est pas trouvée, l'algorithme coupe les intervalles jusqu'à obtenir des tailles d'intervalles inférieures ou égales à la précision.
- ◇ Le nombre de disjonctions présentes dans le CSP augmente le temps de calcul de manière au moins linéaire, voir figure 5.4 page suivante.

5.4 Utilisation d'AbSolute sur le cycle négatif

Dans tous les CSPs issus de la plus faible précondition de la logique de Hoare hybride, nous ajoutons des contraintes spécifiques à la trace biologique :

- ◇ D'une part, les contraintes supplémentaires proviennent du second axiome² présenté en sous-section 4.3.5 page 82. Cet axiome contraint le signe des célérités voisines présentant une même ressource pour une entité donnée :

$$\forall v \in V, \forall \omega \subset R^-(v), \forall k \in \llbracket 0, b_v - 1 \rrbracket, C_{v,\omega,k} \times C_{v,\omega,k+1} \geq 0$$

Ces contraintes sont ajoutées à tous les CSPs provenant d'un calcul de plus faible précondition.

2. Généralement, dans les phénomènes oscillatoires, on n'observe pas de célérité nulle donc le troisième axiome n'a pas été pris en compte dans l'implémentation.

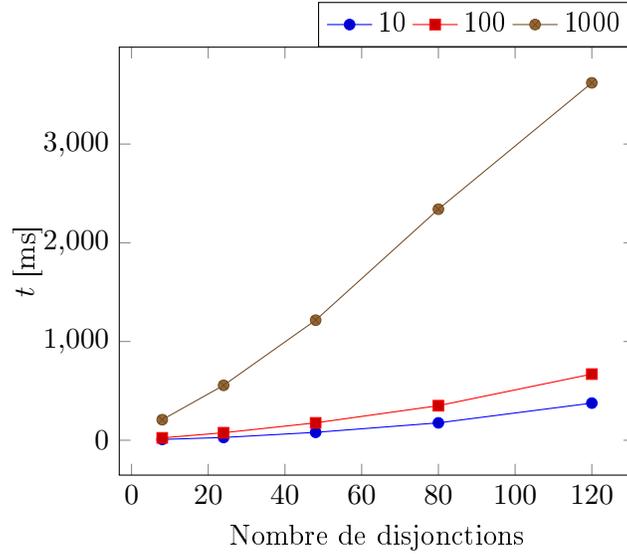


FIGURE 5.4 – Temps de résolution en fonction du nombre de disjonctions dans les contraintes pour un graphe représentant une boucle simple de rétrocontrôle négative. Le temps de résolution augmente lorsque le nombre de disjonctions augmente, quelque soit le nombre de solutions souhaité (10 pour la courbe bleue, 100 pour la courbe rouge et 1000 pour la courbe marron). Plus le nombre de solutions augmente, plus le temps de résolution augmente. La précision pour chaque calcul est de 10^{-3} .

- ◊ D'autre part, des contraintes supplémentaires sont ajoutées si on veut exprimer le caractère cyclique de la trace observée. Dans ce cas, les états hybrides d'entrée en pré et postconditions sont les mêmes : les contraintes rajoutées sont alors $\bigwedge_{v \in V} \pi_v^i = \pi_v^f$, avec i et f les indices des états de pré et postconditions, respectivement. Les contraintes du comportement **cyclique** sont ajoutées uniquement lorsque l'utilisateur spécifie que la trace biologique est cyclique. Lorsque ces contraintes ne sont pas ajoutées, on parle de comportement **acyclique**.

Nous prenons l'exemple du cycle négatif présenté en section 4.6 page 92 où la trace biologique et la postcondition sont également définies. Les contraintes ajoutées aux CSP sont les suivantes :

$$\begin{aligned}
 & \diamond (C_{v_1, \emptyset, 0} \times C_{v_1, \emptyset, 1} \geq 0) \wedge (C_{v_1, \{m_2\}, 0} \times C_{v_1, \{m_2\}, 1} \geq 0) \wedge \\
 & \quad (C_{v_2, \emptyset, 0} \times C_{v_2, \emptyset, 1} \geq 0) \wedge (C_{v_2, \{m_1\}, 0} \times C_{v_2, \{m_1\}, 1} \geq 0), \\
 & \diamond (\pi_{v_1}^0 = \pi_{v_1}^4) \wedge (\pi_{v_2}^0 = \pi_{v_2}^4) \text{ puisque la trace présente un comportement cyclique.}
 \end{aligned}$$

Dans la suite de cette section, nous cherchons des solutions du CSP associées à la trace biologique puis nous donnons quelques résultats d'expériences à propos du temps de résolution pour des traces de plus en plus longues dans des réseaux représentant une boucle de rétrocontrôle négative de plus en plus grande.

5.4.1 Comportement cyclique du cycle négatif

Le CSP du cycle négatif en comportement cyclique est défini par :

- ◊ 26 variables :
 - 10 parties fractionnaires d'entrée π_v^i avec $v = \{v_1, v_2\}$, $i = \llbracket 0, 4 \rrbracket$,
 - 8 parties fractionnaires de sortie π_v^j avec $v = \{v_1, v_2\}$, $j = \llbracket 0, 3 \rrbracket$ (pas de partie fractionnaires de sortie pour l'état satisfaisant la postcondition),
 - 8 célérités dont 4 pour l'entité v_1 et 4 pour l'entité v_2
- ◊ le domaine de définition des parties fractionnaires est $[0, 1]$, celui des célérités est l'ensemble des réels \mathbb{R} ,
- ◊ les 38 contraintes qui sont détaillées en annexe 9.1 page 149.

En considérant une précision entre les bornes des intervalles de 10^{-3} et l'obtention de 10 solutions (sûres ou non sûres), le solveur résout ce CSP en 12 millisecondes et identifie directement une solution sûre. Les domaines de définition des célérités obtenus sont :

$$\begin{aligned}
 D_{C_{v_1, \emptyset, 0}} &= [-0.199999, -0.099999], & D_{C_{v_2, \emptyset, 0}} &=]-\infty, -0.25], \\
 D_{C_{v_1, \{m_2\}, 0}} &= [0.071429, 0.142858], & D_{C_{v_2, \{m_1\}, 1}} &= [0.142858, +\infty[, \\
 D_{C_{v_1, \emptyset, 1}} &= [-0.25, 0.0], & D_{C_{v_2, \{m_1\}, 0}} &= [0.20, 0.20], \\
 D_{C_{v_1, \{m_2\}, 1}} &= [-0.0, 0.125], & D_{C_{v_2, \emptyset, 1}} &= [-0.125, -0.125].
 \end{aligned}$$

Les domaines de définition des parties fractionnaires d'entrée de la précondition sont $D_{\pi_{v_1}^0} = [0.0, 0.0]$ et $D_{\pi_{v_2}^0} = [1.0, 1.0]$. Les domaines des autres parties fractionnaires sont décrites en annexe 9.1, tableau 9.1 page 150. A partir des valeurs obtenues, il est possible de simuler l'évolution du modèle au cours du temps, voir chapitre 6 page 113.

5.4.2 Comportement acyclique du cycle négatif

Le comportement acyclique présente le même CSP que le comportement cyclique, à la différence que les contraintes de position entre pré- et postcondition n'existent plus. Ce nouveau CSP est résolu en 24 millisecondes et ne permet pas d'obtenir de solution sûre. Dans 7 solutions non sûres parmi les 10 obtenues, 5 variables sont invariantes : $C_{v_1, \emptyset, 0}$, $C_{v_1, \{m_2\}, 0}$, $\pi_{v_1}^1$, $\pi_{v_1}^0$ et $\pi_{v_1}^{0'}$. Dans les trois autres solutions, 4 autres variables supplémentaires sont invariantes ; $C_{v_1, \emptyset, 1}$, $C_{v_1, \{m_2\}, 1}$, $\pi_{v_1}^3$ et $\pi_{v_1}^{2'}$. L'appel du solveur ne permet donc pas de donner une solution sûre du CSP dans cet exemple. Néanmoins, nous montrerons dans le chapitre 6 page 113 une heuristique possible pour obtenir au moins une solution lorsque le solveur est incapable de trouver de solutions sûres au CSP.

5.4.3 Comportement cyclique du cycle négatif avec temps Δt inconnus

Reprenons l'exemple du comportement cyclique du cycle négatif. Dans cet exemple, nous considérons que la durée du premier chemin élémentaire est Δt_1 (au lieu de 5.0 heures), que la durée du second chemin élémentaire est Δt_2 (au lieu de 7.0 heures) et que $\Delta t_1 + \Delta t_2 = 12.0$. Ce CSP contient alors des contraintes non linéaires de la forme $\pi_{v_2}^0 = \pi_{v_2}^{0'} - C_{v_2, \{m_1\}, 0} \times \Delta t_1$. Le temps de résolution de ce nouveau CSP est de 26 millisecondes et AbSolute ne donne pas de solution sûre directement, mais démontre qu'il est possible de résoudre des CSPs présentant des contraintes non linéaires selon une heuristique proposée dans le chapitre 6 page 113.

5.4.4 Comportements cycliques de rétrocontrôle négatif de tailles différentes

L'enjeu du dernier exemple traité est d'étudier la dépendance du temps d'exécution du solveur pour résoudre des CSPs en fonction de la taille des graphes d'influence sous-jacents, voir figure 5.5 page suivante. Nous comparons les résultats obtenus pour les 5 CSPs présentés en table 5.1 :

- ◇ le nombre d'entités présentes dans le graphe d'influence varie de 2 à 6 entités,
- ◇ la trace biologique contient entre 4 et 12 chemins élémentaires,
- ◇ le nombre de solutions demandé prend les valeurs 10, 100 et 1000,
- ◇ la précision relative aux bornes d'intervalles change et peut valoir 10^{-3} , 10^{-6} et 10^{-9} .

	a	b	c	d	e
nombre d'entités α	2	3	4	5	6
nombre de chemins élémentaires β	4	6	8	10	12
nombre de célérités γ	8	12	16	20	24
nombre de parties fractionnaires λ	18	39	68	105	150

TABLE 5.1 – Les 5 CSPs utilisés pour évaluer le temps d'exécution du solveur. Le nombre de célérités γ est calculé selon la formule $\sum_v ((b_v + 1) \times 2^{|G^{-1}(v)|})$ avec b_v le niveau d'expression maximal de v et $|G^{-1}(v)|$ le nombre de prédécesseurs de v , tandis que le nombre de parties fractionnaires λ est calculé selon la formule $(2 \times \alpha \times \beta) + \alpha$ (chaque entité présente 2 parties fractionnaires par chemin élémentaire et une partie fractionnaire d'entrée en postcondition).

Ainsi, nous avons comparé 5 graphes d'influence représentant tous une simple boucle de rétroaction négative avec 1 seule influence négative mais dont la taille varie de 2 à 6. Chaque graphe possède une entité inhibitrice et les autres entités sont toutes activatrices. Pour simplifier le calcul, nous considérons que toutes les assertions des chemins élémentaires sont "T", et le temps passé entre chaque événement est fixé aléatoirement (pas d'influence de la valeur de Δt sur le temps de calcul). La figure 5.6 page 111 montre que, pour une précision fixée à 10^{-3} ,

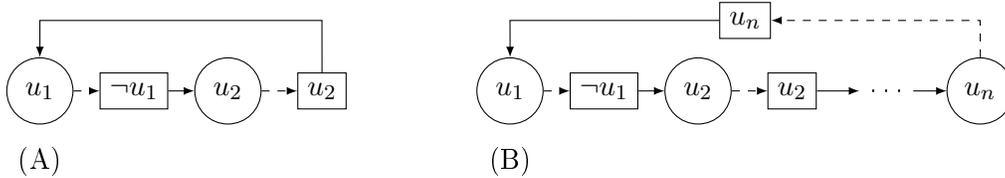


FIGURE 5.5 – Graphe d’influence de boucles négatives. (A) Graphe d’influence de taille 2. (B) Graphe d’influence de taille 3 ou plus. Le multiplexe de u_1 noté $\neg u_1$ est un inhibiteur et les multiplexes u_i avec $i \in \llbracket 2, n \rrbracket$ sont tous des activateurs. Les points \dots représentent une succession d’entités et de leurs multiplexes activateurs correspondants.

le temps de résolution augmente en fonction du nombre d’entités dans le graphe, c’est-à-dire du nombre de variables à identifier (célérités et parties fractionnaires). Il en est de même lorsque le nombre de solutions demandées augmente. Cependant, améliorer la précision entre les bornes des intervalles ne modifie pas significativement le temps de résolution, voir tableau 9.2 page 150.

Les résultats obtenus tendent à montrer que le temps de résolution est exponentiel en fonction du nombre d’entités présentes dans le graphe d’influence, montrant la difficulté de résolution des CSPs provenant de notre logique de Hoare hybride.

En résumé, les CSPs issus de traces biologiques sont définis par :

- ◇ les variables célérités et les parties fractionnaires,
- ◇ leurs domaines de définition ($[0, 1]$ pour les parties fractionnaires et l’ensemble des réels \mathbb{R} pour les célérités),
- ◇ les contraintes provenant du calcul de plus faible précondition.

Nous avons résolu ces CSPs avec le solveur AbSolute sur les réels. L’utilisation des propagateurs réduit les domaines de définition des variables jusqu’à identifier l’ensemble des valeurs possibles qui satisfont le CSP. Nous avons dû modifier quelque peu AbSolute pour que le solveur puisse traiter nos CSPs. La réécriture symbolique et l’ajout des listes d’invariants et de changeants a permis d’identifier des jeux de paramètres pour des boucles négatives simples.

Cependant, il reste des cas où le solveur n’arrive pas à obtenir de solutions sûres du premier coup. Ces situations pourraient être dues à un manque de précision donné par l’utilisateur, à un nombre de disjonctions trop important dans le CSP, ou encore à l’approximation des réels qui peut ne pas satisfaire des contraintes d’égalité alors qu’il le devrait (exemple de la contrainte $x = 5.0/2.0$ où l’approximation des réels pourrait affecter la valeur 2.5000001 à x).

Un prototype, Holmes BioNet³, a été proposé pour rendre l’approche développée dans cette

3. Hoare Logic Modelling Experiments on Biological Networks

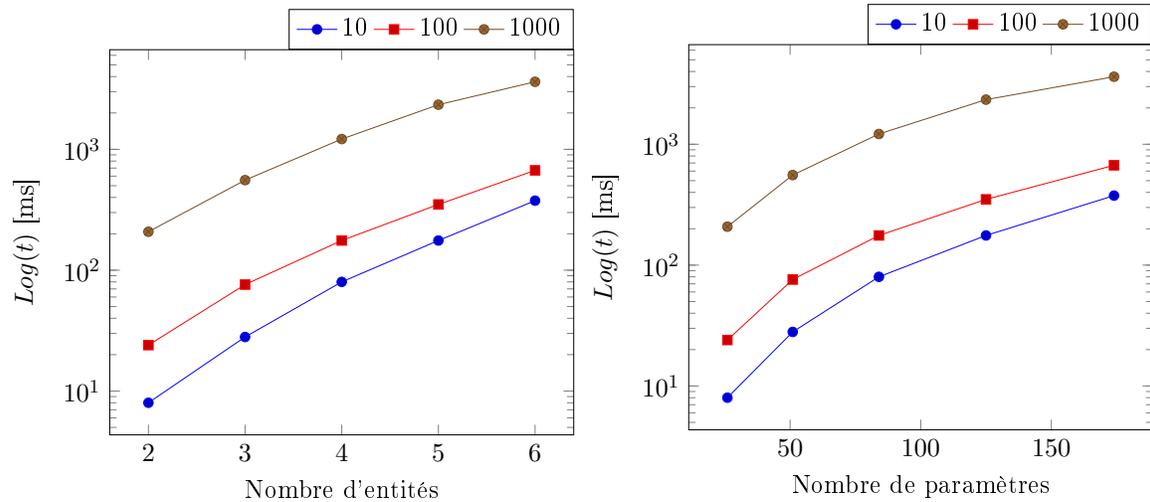


FIGURE 5.6 – Temps de résolution (échelle logarithmique) en fonction du nombre d'entités ou du nombre de paramètres dans un graphe représentant une boucle simple de rétrocontrôle négative. Le temps de résolution augmente lorsque le nombre d'entités ou le nombre de paramètres augmente, quelque soit le nombre de solutions souhaitées (10 pour la courbe bleue, 100 pour la courbe rouge et 1000 pour la courbe marron). Plus le nombre de solutions augmente, plus le temps de résolution augmente. La précision pour chaque calcul est de 10^{-3} .

thèse utilisable. Ce prototype permet de spécifier à la fois le graphe d'influence, la trace biologique observée, utilise le calcul de plus faible précondition de la logique de Hoare hybride, et tente de résoudre le CSP obtenu par des appels successifs au solveur AbSolute. Holmes BioNet est décrit dans le prochain chapitre.

Chapitre 6

Le prototype Holmes BioNet

Dans les chapitres précédents, nous avons d’une part construit les contraintes sur les paramètres dynamiques d’un modèle d’un système biologique, et d’autre part identifié un ou plusieurs jeux de paramètres satisfaisant le CSP. Nous avons conçu le prototype **Holmes BioNet** (pour *HOare Logic Modelling ExperimentS on Biological Networks*) qui se charge de construire des contraintes sur les paramètres dynamiques, d’identifier un ou plusieurs jeux de paramètres, et de simuler l’évolution du modèle dans le but de la comparer avec les traces expérimentales. A partir d’un graphe d’influence et d’une trace biologique donnés, Holmes BioNet effectue les 4 étapes suivantes, voir figure 6.1 page suivante :

- ◊ Analyse des données d’entrée et construction du graphe d’influence et de la trace biologique donnés,
- ◊ Construction des contraintes sur les paramètres dynamiques,
- ◊ Identification d’un ou plusieurs jeux de paramètres satisfaisant le CSP,
- ◊ Simulation de la dynamique du modèle à partir d’un jeu de paramètres choisi.

Finalement, ce chapitre illustre ces différentes étapes à l’aide du graphe d’influence du cycle circadien avec multiplexes présenté en figure 6.2 page suivante et dont les mécanismes d’action ont été présentés en figure 2.11 page 39. Pour rappel, ce graphe contient 3 entités : les complexes protéiques BMAL1/CLOCK (noté BC), PER/CRY (noté P) et la protéine REV-ERB α (noté R). BC est un activateur de R et P , R est un inhibiteur de BC et P , puis P joue le rôle d’inhibiteur de BC .

6.1 Données d’entrée : Graphe d’influence et trace biologique

Holmes BioNet récupère les informations dont il a besoin dans un fichier d’entrée décrivant le graphe d’influence et la trace biologique. Ces informations proviennent de la littérature et

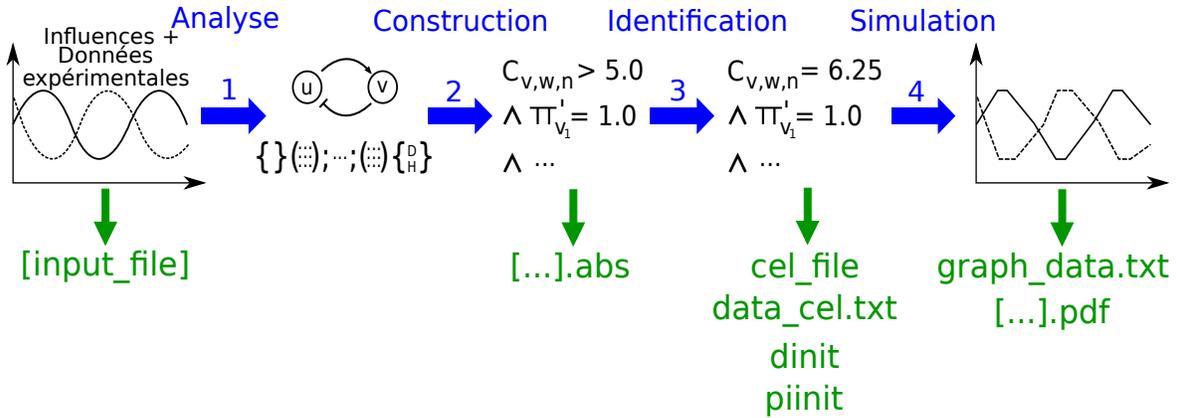


FIGURE 6.1 – Pipeline du prototype. Les 4 étapes du prototype Holmes BioNet sont indiquées en bleu et correspondent à l'analyse des données d'entrée menant à construire le graphe d'influence et la trace biologique donnés (1), la construction des contraintes sur les paramètres dynamiques (2), l'identification d'un ou plusieurs jeux de paramètres (3) et la simulation de la dynamique à partir d'un jeu de paramètres choisi (4). Les fichiers indiqués en vert à chaque étape correspondent aux différents fichiers créés par Holmes BioNet, excepté le fichier `[input_file]` créé par l'utilisateur et analysé par Holmes BioNet pour construire la forme interne du graphe d'influence et la trace biologique selon les définitions des chapitres 3 et 4.

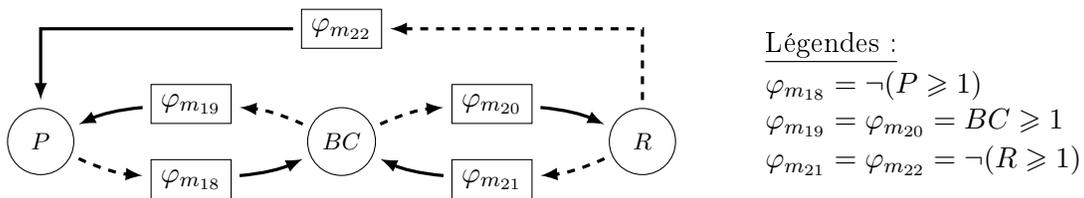


FIGURE 6.2 – Graphe d'influence de l'horloge circadienne à 3 entités avec multiplexes. Ce graphe, similaire au graphe de la figure 2.11 page 39, utilise des multiplexes pour représenter les régulations entre les entités du graphe.

des observations expérimentales de l'utilisateur. Le fichier d'entrée de Holmes BioNet contient deux parties qui définissent le graphe d'influence et la trace biologique.

6.1.1 Ecriture du graphe d'influence dans le fichier d'entrée

La représentation d'un graphe d'influence nécessite la définition des entités et des multiplexes du graphe. L'exemple suivant illustre la syntaxe utilisée pour représenter le graphe d'influence :

```
1 Start Influence Graph
2 var P 1;
3 var BC 1;
4 var R 1;
5
6 mult m1
7   formula: Neg(P >= 1)
8   targets: BC;
9
10 mult m2
11   formula: BC >= 1
12   targets: P;
13
14 mult m3
15   formula: BC >= 1
16   targets: R;
17
18 mult m4
19   formula: Neg(R >= 1)
20   targets: BC;
21
22 mult m5
23   formula: Neg(R >= 1)
24   targets: P;
25 End Influence Graph
```

Le graphe d'influence est décrit entre les phrases "Start Influence Graph" et "End Influence Graph". Une variable¹ se définit par une phrase en commençant par le terme "var" suivi du nom de la variable, de son niveau d'expression maximal et du caractère final ";". Par exemple, la ligne "var BC 1;" signifie que l'entité *BC* est définie et que son niveau d'expression maximal est $b_{BC} = 1$.

Ensuite, chacun des multiplexes est défini sur 3 lignes de la façon suivante :

1. Les entités dans le fichier sont des variables, d'où le terme "var" employé. Quand on parle spécifiquement du graphe d'influence, on parlera d'entités.

Atome A <i>comp</i> B	Syntaxe	Formule logique F	Syntaxe
$A > B$	A>B	$F_1 \wedge F_2$	F1 And F2
$A < B$	A<B	$F_1 \vee F_2$	F1 Or F2
$A \geq B$	A>=B	$\neg F_1$	Neg(F1)
$A \leq B$	A<=B		

TABLE 6.1 – Syntaxe des formules de multiplexe (à gauche la syntaxe pour les atomes comparants 2 termes, à droite la syntaxe pour les connecteurs logiques).

- ◊ la première ligne commence par le terme "mult" suivi du nom du multiplexe (exemple : "mult m1"²),
- ◊ la seconde ligne décrit la formule associée au multiplexe en commençant par l'expression "formula:". La formule utilise la syntaxe décrite dans le tableau 6.1,
- ◊ la dernière ligne définit les cibles du multiplexe et la ligne commence par l'expression "targets:" suivi du nom des entités cibles séparées par une virgule s'il y en a plusieurs, et termine par le caractère ";".

Par exemple, le multiplexe $m_1 = \neg(P \geq 1)$ ciblant l'entité BC est défini de la manière suivante :

```
mult m1
  formula: Neg(P >= 1)
  targets: BC;
```

Ce multiplexe indique alors que l'entité P est un inhibiteur de BC .

6.1.2 Ecriture du triplet de Hoare dans le fichier d'entrée

La représentation d'un triplet de Hoare issu d'observations expérimentales dans le fichier peut être illustré à l'aide de l'exemple suivant :

```
Start Hoare Triple
{}
(6.12,Slide-(R),P-);
(3.44,True,BC+);
(2.44,Slide+(BC) And Slide-(P),R+);
(6.12,Slide+(R),P+);
(3.44,True,BC-);
(2.44,Slide-(BC) And Slide+(P),R-)
```

². le caractère _ est interdit dans le fichier d'entrée. Par souci de lisibilité, nous considérons que m_1 signifie m_1 .

```
{
eta(P) = 1 And Eta(R) = 0 And Eta(BC) = 0,
True
}
```

End Hoare Triple

Le triplet de Hoare est décrit entre les lignes "Start Hoare Triple" et "End Hoare Triple". Puisque nous identifions les contraintes minimales de précondition, il n'est pas nécessaire d'écrire des informations pour ce prédicat. Néanmoins, nous avons fait le choix de représenter la précondition par la ligne "{" pour visualiser facilement un triplet de Hoare dans notre fichier d'entrée.

La trace biologique est ensuite représentée par une succession de chemins élémentaires. Chaque chemin élémentaire est une ligne de la forme "(T,a,dpa);" où "T" est le temps passé dans l'état, "a" représente les symboles d'assertions et "dpa" correspond à l'atome de chemin discret. Notons que le dernier chemin élémentaire ne présente pas le caractère ";" en fin de ligne. Les éléments de chaque chemin élémentaire sont définis de la manière suivante :

- ◊ le temps passé dans l'état Δt représenté ici par "T" est une valeur positive réelle (exemple : "6.12"),
- ◊ les symboles d'assertions utilisés sont indiqués dans le tableau 6.2 (exemple de symboles d'assertions : "Slide-(R)"),
- ◊ l'atome de chemin discret représenté par "dpa" est la concaténation d'une variable précédemment définie avec le signe "+" ou "-" selon l'incréméntation ou la décrémentation du niveau d'expression de l'entité correspondante (exemple : "P-").

Ainsi, le chemin élémentaire $\begin{pmatrix} 6.12 \\ \text{slide}^-(R) \\ P- \end{pmatrix}$ est représenté dans le fichier d'entrée par la ligne "(6.12,Slide-(R),P-)" concaténée avec le caractère ";" si ce n'est pas le dernier chemin élémentaire de la trace biologique.

Assertion a	Syntaxe	Assertion a	Syntaxe
\top	True	$\text{slide}(v)$	Slide(v)
$C_v \text{ comp } c$	Cv comp c	$\text{slide}^+(v)$	Slide+(v)
$\neg a$	Neg(a)	$\text{slide}^-(v)$	Slide-(v)
$a_1 \wedge a_2$	a1 And a2	$\text{noslide}(v)$	NoSlide(v)
$a_1 \vee a_2$	a1 Or a2	$\text{noslide}^+(v)$	NoSlide+(v)
		$\text{noslide}^-(v)$	NoSlide-(v)

TABLE 6.2 – Syntaxe du langage d'assertion pour le fichier d'entrée. Le terme comp représente les comparateurs et sont définis comme indiqué dans le tableau 6.1 page précédente.

La postcondition est aussi représentée dans le fichier selon la syntaxe " $\{D, H\}$ " où " D " est la condition discrète et " H " est la condition hybride en suivant la syntaxe du tableau 6.3. Par exemple, si l'état qualitatif de postcondition est connu $((\eta_P, \eta_R, \eta_{BC}) = (1, 0, 0))$ mais il n'y a aucune information concernant la condition hybride, la postcondition s'écrit " $\{\text{eta}(P) = 1 \text{ And } \text{Eta}(R) = 0 \text{ And } \text{Eta}(BC) = 0, \text{True}\}$ " (" True " signifiant qu'il n'y a aucune information).

Terme discret td	Syntaxe
η_v	$\text{Eta}(v)$
c (constante sur \mathbb{N})	c

Condition discrète D	Syntaxe
$td_1 \text{ opérateur } td_2$	$td1 \text{ opérateur } td2$
$\neg D$	$\text{Neg}(D)$
$D_1 \wedge D_2$	$D1 \text{ And } D2$
$D_1 \vee D_2$	$D1 \text{ Or } D2$

Terme continu a_c	Syntaxe
π_v	$\text{Pi}(v)$
π'_v	$\text{Pi}'(v)$
$C_{v, \{m_1, m_2\}, k}$	$\text{C}(v, [m1, m2], k)$
n ($\in \llbracket 0, b_v \rrbracket$)	n
c (constante sur \mathbb{R})	c

Condition hybride H	Syntaxe
$td_1 \text{ opérateur } td_2$	$td1 \text{ opérateur } td2$
$tc_1 \text{ opérateur } tc_2$	$tc1 \text{ opérateur } tc2$
$\neg H$	$\text{Neg}(H)$
$H_1 \wedge H_2$	$H1 \text{ And } H2$
$H_1 \vee H_2$	$H1 \text{ Or } H2$

TABLE 6.3 – Syntaxe concrète du langage de propriétés pour le fichier d'entrée. Le terme *opérateur* représente les connecteurs $+$, $-$, $*$ et \setminus pour créer un nouveau terme. Le connecteur \setminus est utilisé uniquement pour les termes continus.

Pour compléter le triplet de Hoare dans le fichier d'entrée, il est possible d'indiquer que la trace biologique est cyclique. Cette information s'écrit à la fin du fichier par la ligne "**Cyclic behaviour**" et indique des égalités entre les parties fractionnaires d'entrée des pré et postconditions de chaque entité.

Le fichier d'entrée ainsi créé va être lu par Holmes BioNet : ce fichier est parsé (autrement dit la syntaxe est analysée) pour construire la forme interne du graphe d'influence et du triplet de Hoare selon les définitions présentées dans les chapitres 3 et 4. Il est alors possible d'effectuer le calcul de plus faible précondition pour construire les contraintes minimales sur les paramètres dynamiques du système.

6.2 Construction des contraintes

Holmes BioNet construit les contraintes en 4 étapes :

1 - La première étape consiste à identifier les états qualitatifs parcourus par la trace biologique. Cela permet d'une part de vérifier que la trace discrète est compatible avec l'espace des états discrets et avec la condition discrète de la postcondition, et d'autre part d'identifier les ressources des entités dans chaque état qualitatif parcouru. En effet, la connaissance des niveaux d'expression des entités permet d'évaluer les formules de chacun des multiplexes. Nous pouvons alors identifier les célérités utilisées dans chaque état qualitatif parcouru.

2 - La seconde étape consiste à calculer la plus faible précondition à partir de la stratégie *backward*. En partant de la postcondition, chaque chemin élémentaire parcouru mène à construire progressivement la précondition selon la définition 4.7 page 81. Le *dpa* ainsi que la connaissance du temps passé Δt dans le chemin élémentaire permettent de contraindre spécifiquement les célérités dans l'état courant. De plus, les symboles de l'assertion *a* sont traduits et permettent de contraindre davantage les célérités de l'état courant comme présenté dans la sous-section 4.3.4 page 78.

3 - La troisième étape consiste à simplifier et réduire les contraintes construites en utilisant les lois de De Morgan. En effet, les contraintes identifiées contiennent de nombreuses sous-formules qu'il est possible de simplifier ce qui va drastiquement faciliter le travail du solveur de contraintes par la suite puisque cette simplification va réduire le nombre de clauses disjonctives.

4 - La dernière étape consiste à écrire les contraintes dans un fichier de sortie (fichier *.abs*) qui sera lu par le solveur AbSolute. Ce fichier contient 3 parties :

- ◇ la première partie définit les constantes en commençant par le terme "**constants**". L'ensemble des constantes est défini entre des accolades et chaque constante est représentée sur une ligne de la forme "**var=value;**" où "**var**" est une partie fractionnaire ou une célérité représentée selon la syntaxe indiquée dans le tableau 6.4 page suivante et "**value**" est la valeur réelle associée à cette variable. Par exemple, si $\pi_R' = 0.0$ et $\pi_{BC}' = 0.0$, alors les constantes s'écrivent "**constants{pi_exit_R_1=0.;pi_exit_BC_1=0.;}**".
- ◇ la deuxième partie définit les domaines de définition des variables. Ces domaines sont définis entre des accolades et sont précédées par le terme "**init**". Chaque domaine est représenté sur une ligne de la forme "**real var = [min;max];**" où "**var**" est la variable, "**min**" est la valeur minimale et "**max**" est la valeur maximale (voir tableau 6.4 page suivante). Les parties fractionnaires sont définies sur $[0;1]$ et les célérités sur $[-50;50]$ ³ (exemple : "**init{real pi_entrance_R_0 = [0;1];real C_R_m3_m4_0 = [-50;50];}**").
- ◇ la troisième partie définit les contraintes du système entre accolades, précédées par le terme "**constraints**" suivi de la contrainte selon la syntaxe des variables, des opérateurs et des connecteurs définis précédemment. Par exemple, les contraintes $\pi_P^0 =$

3. Nous avons borné sur $[-50, 50]$ plutôt que $]-\infty; \infty[$ pour faciliter la résolution des contraintes.

π_P^6 (comportement cyclique) et $C_{P,\emptyset,0} \times C_{P,\emptyset,1} \geq 0$ sont définies dans le fichier par "constraints{pi_entrance_P_0=pi_entrance_P_6;C_P____0 * C_P____1>=0.;}".

Variables	Syntaxe
π_v^i	pi_entrance_v_i
$\pi_v^{i'}$	pi_exit_v_i
$C_{v,\emptyset,n}$	C_v____n
$C_{v,\{m_1\},n}$	C_v__m1__n
$C_{v,\{m_1,m_2\},n}$	C_v__m1_m2__n

TABLE 6.4 – Syntaxe des célérités et des parties fractionnaires pour le fichier de sortie qui sera lu par le solveur AbSolute. v représente une entité, n son niveau d’expression, m_1 et m_2 les noms des multiplexes de v et i l’indice de l’état qualitatif courant.

Le fichier ainsi créé représente le CSP associé à la trace biologique et va être transmis au solveur AbSolute.

6.3 Appel au solveur AbSolute

Le solveur sur les réels AbSolute est appelé et permet de résoudre le CSP décrit dans le fichier créé précédemment. Les solutions retournées sont, pour toutes les variables, soit des valeurs réelles, soit des intervalles. Pour rappel, les variables de la trace biologique à identifier sont les parties fractionnaires et les célérités.

Lorsque le solveur est appelé, il résout le système de contraintes en réalisant une étape de réécriture symbolique et identifie les domaines de définition invariants et changeants comme indiqué dans le chapitre 5 page 97. L’objectif du solveur consiste alors à trouver l’ensemble des solutions (sûres et non sûres).

Lorsque l’appel du solveur ne permet pas de donner une solution sûre du CSP, nous choisissons au hasard un domaine de définition changeant et nous fixons la valeur de cette variable aléatoirement dans son domaine de définition. Cette nouvelle constante est intégrée dans le fichier d’entrée et le solveur est de nouveau appelé avec ce fichier modifié. Nous réitérons ce procédé tant qu’une solution sûre n’est pas exhibée ou tant qu’aucune solution ne soit trouvée. Dès lors que toutes les célérités sont identifiées, elles sont écrites dans le fichier "cel_file".

Ainsi nous obtenons au moins un jeu de paramètres qui permet de simuler le comportement des entités du graphe d’influence au cours du temps. C’est une approche heuristique qui permet ainsi de confronter le paramétrage obtenu par Holmes BioNet et les traces qui en découlent avec les données expérimentales.

6.4 Simulateur

Le simulateur de Holmes BioNet permet de calculer les niveaux d'expression hybrides des entités au cours du temps pour un jeu de paramètres choisi. Le niveau d'expression hybride d'une entité est la somme de son niveau qualitatif et de sa partie fractionnaire.

Réaliser cette simulation nécessite de connaître un état qualitatif initial et ses parties fractionnaires, ainsi que tous les paramètres dynamiques du système biologique. La durée de la simulation est fixée par l'utilisateur et sa valeur par défaut est de 72 unités de temps.

Pré-traitement Dans le fichier "`cel_file`", lorsque le domaine de définition d'une célérité est un intervalle, nous fixons au hasard cette variable par une valeur dans son domaine de définition. Au final, le jeu de paramètres identifié est écrit dans le fichier "`data_cel.txt`".

De plus, réaliser une simulation requiert de connaître le point initial. Puisque toutes les parties fractionnaires sont connues ainsi que les niveaux d'expression pour chaque chemin élémentaire de la trace biologique, nous fixons le point initial comme étant l'état hybride de la précondition obtenue. Cet état initial est écrit sous forme de deux fichiers que sont "`dinit`" pour les niveaux qualitatifs des entités et "`piinit`" pour leur parties fractionnaires.

Simulation Le simulateur traite les cas particuliers du cadre de modélisation hybride :

- ◊ lorsqu'un état d'équilibre est atteint avant la fin de la simulation, les niveaux d'expression des entités n'évoluent plus et le simulateur s'arrête,
- ◊ lorsque toutes les célérités des entités sont nulles, les délais de contacts deviennent infinis.

Il est donc impossible d'atteindre un bord d'une entité et de le franchir.

Post-traitement Le simulateur crée alors deux fichiers, le premier est un fichier texte indiquant le niveau d'expression (somme du niveau qualitatif et de la partie fractionnaire) de chaque entité au cours du temps, tandis que le second est son graphique (fichier pdf généré par l'outil graphique Gnuplot en ligne de commande) nommé "`result_[input_file]_[time]H.pdf`" où "`[input_file]`" est le nom du fichier d'entrée et "`[time]`" est le temps maximal simulé (et non le temps d'exécution).

Dans le fichier texte, chaque ligne correspond à un temps précis au cours de la simulation et les niveaux d'expression hybrides sont écrits en séparant les colonnes par des tabulations. La première ligne donne la signification de chaque colonne. Cette ligne commence toujours par le terme "`Time`" suivi des noms des variables. Par exemple, la ligne "`Time P BC R`" indique que le niveau d'expression de l'entité P sera en deuxième colonne de chaque ligne, celui de BC en troisième colonne et celui de R en quatrième colonne.

Chapitre 6 : Le prototype Holmes BioNet

Chaque ligne suivante présente en première colonne le temps où un événement a lieu, suivi des niveaux d'expression de toutes les entités (valeurs réelles correspondant à la somme du niveau qualitatif de l'entité η et de sa partie fractionnaire π). Le résultat suivant est un extrait du fichier obtenu pour le graphe d'influence du cycle circadien :

```
1 Time P BC R
2 0. 2. 9.57361220485e-13 1.
3 2.55574731537e-14 2. 0. 0.999999999999
4 0.039869707391 1.99348536955 0. 0.
5 6.12002594891 1. 0. 0.
6 6.12002594891 1. 0. 0.
7 6.14669261558 0. 0.00775194666667 0.
```

Cet extrait montre au temps $t = 0$ ligne 2 que les niveaux d'expression hybrides de P , BC et R sont respectivement 2.0, $9.57361220485e-13$ et 1.0. Ces valeurs correspondent à l'état qualitatif $(\eta_P, \eta_{BC}, \eta_R) = (1, 0, 0)$ et les parties fractionnaires $(\pi_P, \pi_{BC}, \pi_R) = (1.0, 9.57361220485e-13, 1.0)$.

La ligne 3 montre que l'entité BC atteint son bord inférieur (valeur 0.0) et glisse car sa valeur reste inchangée jusqu'à la ligne 6. L'évolution des niveaux d'expression des entités se poursuit (ligne 4) et l'entité R atteint également son bord inférieur (valeur 0.) et fait face à un mur externe car son niveau d'expression est 0. Ensuite, P atteint son bord inférieur en ligne 5 et le franchit pour atteindre un nouvel état qualitatif (ligne 6) : les niveaux d'expression hybrides sont donc inchangés entre la ligne 5 et 6 mais le niveau qualitatif de P passe de 1 à 0 et sa partie fractionnaire passe de 0.0 à 1.0. Enfin, P atteint son bord inférieur et fait face à un mur externe (ligne 7).

A partir de tels fichiers, nous utilisons l'outil Gnuplot pour visualiser l'évolution au cours du temps des niveaux d'expression des entités, voir section 6.5 page suivante. Le fichier texte est lu puis interprété pour représenter les coordonnées des entités à chaque temps indiqué dans la première colonne. De plus, puisque les entités peuvent correspondre au jour (*day*) ou à la nuit (*night*), nous détectons ces entités et les représentons selon la couleur en arrière-plan du graphique : grise pour la nuit (activité de *night* supérieure ou égale à 1) et blanche pour la journée (activité de *day* supérieure ou égale à 1), voir annexe 9.3 page 151 présentant un graphe d'influence de la synthèse des protéines PER et CRY en fonction du rythme nyctéméral.

Le graphique obtenu pourra ainsi être comparé aux observations expérimentales.

6.5 Lancement du prototype et résultats

Pour utiliser le prototype Holmes BioNet, le chemin absolu du solveur AbSolute doit être spécifié dans le fichier de configuration *settings.conf*. Dans le répertoire du prototype, la commande à lancer est `./modelling.sh -i FILE [OPTIONS]` où "FILE" correspond au fichier d'entrée représentant le graphe d'influence et le triplet de Hoare défini en section 6.1 page 113, et "OPTIONS" est pris parmi les éléments suivants :

- ◇ `-o, --output` : déclare le chemin et le nom du fichier de sortie (Défaut : le fichier est nommé `input_filename.abs` où "input_filename" est le nom du fichier d'entrée du graphe d'influence et du triplet de Hoare, et il est créé dans un répertoire nommé à la date du jour dans le répertoire courant),
- ◇ `-n, --number` : indique le nombre de solutions demandées au solveur AbSolute (Défaut : 10),
- ◇ `-t, --time` : donne le temps simulé maximum (en heures) pour le modèle (et non un temps d'exécution) dont la valeur par défaut est 72 unités de temps,
- ◇ `-p, --precision` : indique la précision du solveur (Défaut : 10^{-3}),
- ◇ `-h, --help` : affiche la liste des options.

Si une option n'est pas indiquée dans la commande, la valeur par défaut est automatiquement intégrée au programme. Les informations par défaut des différentes options sont indiquées dans le fichier *settings.conf* et peuvent être modifiées.

Cette commande permet donc de construire un fichier contenant les contraintes des paramètres dynamiques du système (fichier ".abs"), d'identifier un jeu de paramètres pour les célérités (fichier "data_cel.txt") et de réaliser une simulation menant au fichier de sortie ("graph_data.txt") et à son graphique ("result_[input_file]_[time]H.pdf").

6.5.1 Cycle circadien

Pour comprendre la dynamique du graphe d'influence avec 3 entités du cycle circadien, nous nous sommes appuyés sur des résultats expérimentaux dans le foie de la souris et qui montrent l'évolution de la fixation des protéines BMAL1 [109], CLOCK et PER/CRY [80], et REV-ERB α [27] sur leur gènes cibles. La méthode utilisée dans ces trois articles consiste à analyser l'ensemble des sites de fixation des facteurs de transcription d'intérêt en utilisant la méthode ChIP-Seq qui combine l'immunoprécipitation de la chromatine et le séquençage ADN. Ce type de donnée a été préféré aux profils d'expression des ARNs car ces données reflètent bien l'activité des facteurs de transcription. Les interactions entre protéines et ADN sont analysées et les anticorps s'associent aux protéines cibles pour les récupérer, puis l'ADN lié à cette protéine cible est relâché et séquencé. Les protéines liées aux gènes *bmal1*, *clock*, *per*, *cry* et *rev-erba* sont

analysées au cours du temps et permettent de déterminer quand ces gènes sont actifs. Cette analyse nous a mené à construire la trace biologique et la postcondition suivantes :

$$\begin{pmatrix} 6.12 \\ \text{slide}^-(R) \\ P- \end{pmatrix}; \begin{pmatrix} 3.44 \\ \top \\ BC+ \end{pmatrix}; \begin{pmatrix} 2.44 \\ \text{slide}^+(BC) \wedge \text{slide}^-(P) \\ R+ \end{pmatrix}; \begin{pmatrix} 6.12 \\ \text{slide}^+(R) \\ P+ \end{pmatrix}; \\ \begin{pmatrix} 3.44 \\ \top \\ BC- \end{pmatrix}; \begin{pmatrix} 2.44 \\ \text{slide}^-(BC) \wedge \text{slide}^+(P) \\ R- \end{pmatrix} \left\{ \begin{array}{c} (\eta_P = 1) \wedge (\eta_R = 0) \wedge (\eta_{BC} = 0) \\ \top \end{array} \right\}$$

La trace biologique présente 6 chemins élémentaires d'une durée totale de 24 heures ($2 \times 6.12 + 2 \times 3.44 + 2 \times 2.44$). Le premier chemin élémentaire indique que le complexe PER/CRY (entité P) devient inactif ($P-$) dans 6.12 heures. Avant cette inactivité, la protéine REV-ERB α est totalement dégradée ($\text{slide}^-(R)$). Le chemin élémentaire suivant informe de l'activation du complexe BMAL1/CLOCK ($BC+$) suivi de REV-ERB α ($R+$) et PER/CRY ($P+$), et ces protéines deviennent successivement inactives ($BC-$ puis $R-$ et de nouveau $P-$ car le système est cyclique). Ces informations sont en accord avec les connaissances biologiques présentées dans le chapitre 2.

Le graphe d'influence et les informations précédentes ont été écrits dans un fichier nommé *path_circadian_clock* dans le répertoire `./HolmesBioNet/examples/`. Nous avons également indiqué que le comportement attendu est cyclique avec la ligne `"Cyclic behaviour"`.

A partir de ce fichier, nous avons lancé la commande suivante :

```
./modeling.sh -i ./HolmesBioNet/examples/path_circadian_clock -t 48
```

Le fichier créé pour le solveur indique que 20 célérités et 42 parties fractionnaires sont définies à partir de cette trace biologique. L'appel du solveur a permis d'identifier un jeu de paramètres et une simulation a été réalisée. Le graphique obtenu est présenté en figure 6.3 page ci-contre et le fichier `"graph_data.txt"` est présenté sous forme de tableau en annexe 9.4 page 152. Ces résultats montrent un comportement cyclique des 3 entités sur une période de 24 heures. Pour

le premier chemin élémentaire $\begin{pmatrix} 6.12 \\ \text{slide}^-(R) \\ P- \end{pmatrix}$, on peut remarquer sur le graphique que l'entité R atteint son bord inférieur (courbe bleue) et glisse ($\text{slide}^-(R)$) avant 6.12 heures, puis l'entité P atteint son bord inférieur (courbe violette) et le franchit à 6.12 heures. Le résultat obtenu est donc en accord avec le chemin élémentaire. Il en est de même pour les 5 chemins élémentaires suivants où les atomes de chemin discret franchissent leur bord au moment attendu, et les glissements définis sont tous observés. La trace exhibée est donc cohérente avec les observations

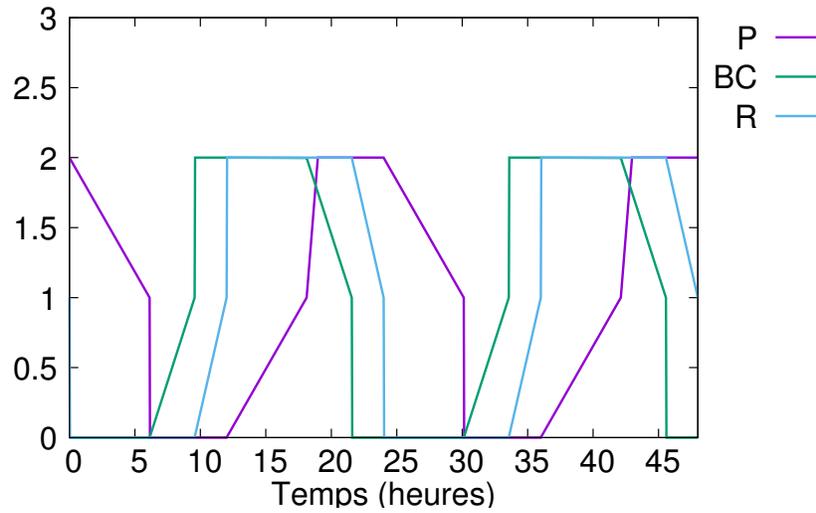


FIGURE 6.3 – Evolution des 3 entités du cycle circadien au cours du temps. L'axe des ordonnées correspond au niveau d'expression hybride des entités, autrement dit à $\eta_v + \pi_v$ pour chaque entité v du système.

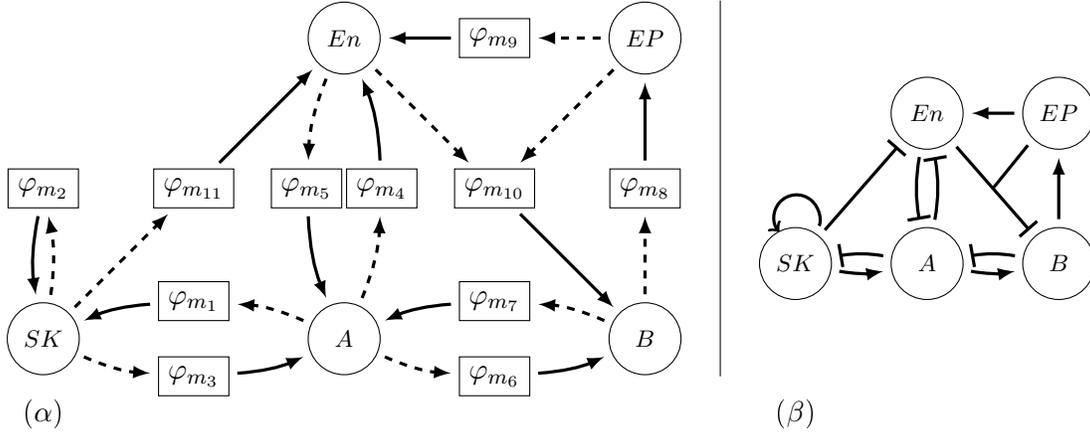
expérimentales données.

6.5.2 Cycle cellulaire

Le graphe d'influence du cycle cellulaire sur lequel nous nous basons est présenté en figure 2.10 page 37. Pour s'adapter au cadre de modélisation hybride, nous transformons ce graphe pour y inclure les multiplexes (au nombre de 11) comme détaillé en figure 6.4 page suivante. Notons que l'entité SK est multivaluée car elle présente deux influences qui n'ont pas lieu au même seuil : le complexe CAK favorise la synthèse de la protéine $CycE$ (lorsque $SK \geq 1$), puis $CycE$ favorise la synthèse de $CycA$ (lorsque $SK \geq 2$).

La construction d'une trace biologique du cycle cellulaire prend en compte les connaissances biologiques présentées dans le chapitre 2 page 19, complétées avec les informations suivantes :

- ◊ d'après les travaux de Feillet *et al.* [47], la durée du cycle cellulaire est d'environ 22.5 heures où :
 - la phase $G1$ dure 10 heures,
 - la phase S dure 8 heures,
 - la phase $G2$ dure 4 heures,
 - la phase M dure 0.5 heures.
- ◊ il y a 12 événements biologiques et donc 12 chemins élémentaires (3 pour $G1$, 4 pour S , 2 pour $G2$ et 3 pour M),
- ◊ les événements importants des phases sont détaillés dans le tableau 6.5 page 127.



Légendes :

$$\begin{array}{lll}
 \varphi_{m_1} = \neg(A \geq 1) & \varphi_{m_4} = \neg(A \geq 1) & \varphi_{m_7} = \neg(B \geq 1) \\
 \varphi_{m_2} = SK \geq 1 & \varphi_{m_5} = \neg(En \geq 1) & \varphi_{m_8} = B \geq 1 \\
 \varphi_{m_3} = SK \geq 2 & \varphi_{m_6} = A \geq 1 & \varphi_{m_9} = EP \geq 1 \\
 \varphi_{m_{10}} = \neg(En \geq 1 \vee (EP \geq 1 \wedge En \geq 1)) & & \varphi_{m_{11}} = \neg(SK \geq 1)
 \end{array}$$

FIGURE 6.4 - 2 graphes d'influence du cycle cellulaire avec 5 entités. Le graphe (β) est une version simplifiée du graphe (α). Les protéines présentes dans les 5 entités sont détaillées en figure 2.10 page 37.

Cependant, bien que la durée de chaque phase est connue, nous ne connaissons pas le temps précis de chaque événement à l'intérieur de ces phases. Nous avons alors émis l'hypothèse de répartir uniformément le temps de chaque phase entre ses chemins élémentaires (3.33 heures pour les chemins élémentaires de $G1$, 2 heures pour ceux de S et $G2$, et 0.17 heures pour ceux de M).

De plus, nous avons construit la postcondition où la condition discrète est identifiée de façon intuitive par simple analyse des atomes de chemin discret de la trace biologique.

Ainsi, nous obtenons la trace biologique et la postcondition suivantes⁴ :

$$\underbrace{\left(\begin{array}{c} 3.33 \\ \text{slide}^-(EP) \\ SK+ \end{array} \right); \left(\begin{array}{c} 3.33 \\ \top \\ SK+ \end{array} \right); \left(\begin{array}{c} 3.33 \\ \text{slide}^+(SK) \\ En- \end{array} \right)}_{G1}$$

4. Des parenthèses autour des slide dans le chemin élémentaire et des η en postcondition sont ajoutées pour une meilleure lisibilité.

Entité	Observations
<i>SK</i>	<i>SK</i> augmente au début de la phase <i>G1</i>
	Activation totale de CycE par CAK
	<i>SK</i> diminue dès l'apparition de <i>A</i>
	Dégradation totale de <i>SK</i> avant l'activation de <i>B</i>
<i>EP</i>	<i>EP</i> est active lorsque <i>B</i> est présente
	<i>EP</i> agit lors du mécanisme de surveillance mitotique
	<i>EP</i> diminue lorsque <i>B</i> est inactive
	Le cycle est dérégulé si l'activité de <i>EP</i> est maintenue
<i>A</i>	<i>A</i> est active au début de la phase <i>S</i>
	<i>A</i> est saturée quand <i>SK</i> est active
	<i>A</i> est dégradée quand <i>B</i> est active
	<i>A</i> est totalement dégradée en début de phase <i>M</i>
<i>B</i>	<i>B</i> est active au début de la phase <i>S</i>
	<i>B</i> est saturée en début de phase <i>M</i>
	<i>B</i> diminue dès l'activation de <i>EP</i> et <i>En</i>
	<i>B</i> est dégradée totalement en fin de phase <i>M</i>
<i>En</i>	<i>En</i> est active quand <i>EP</i> est active
	<i>En</i> est saturée durant la phase <i>M</i>
	<i>En</i> est inhibée quand <i>SK</i> est active
	<i>En</i> est totalement inhibée en début de phase <i>S</i>

TABLE 6.5 – Evénements de début et de fin des phases du cycle cellulaire.

$$\underbrace{\left(\begin{array}{c} 2.0 \\ \text{slide}^-(En) \\ A+ \end{array} \right); \left(\begin{array}{c} 2.0 \\ \top \\ SK- \end{array} \right); \left(\begin{array}{c} 2.0 \\ \text{slide}^+(A) \\ SK- \end{array} \right); \left(\begin{array}{c} 2.0 \\ \text{slide}^-(SK) \\ B+ \end{array} \right)}_S$$

$$\underbrace{\left(\begin{array}{c} 2.0 \\ \top \\ A- \end{array} \right); \left(\begin{array}{c} 2.0 \\ \text{slide}^+(B) \\ EP+ \end{array} \right)}_{G2} \underbrace{\left(\begin{array}{c} 0.17 \\ \text{slide}^+(EP) \\ En+ \end{array} \right); \left(\begin{array}{c} 0.17 \\ \text{slide}^-(A) \wedge \text{slide}^+En \\ B- \end{array} \right); \left(\begin{array}{c} 0.17 \\ \text{slide}^-(B) \\ EP- \end{array} \right)}_M$$

$$\left\{ \left(\begin{array}{c} (\eta_{SK} = 0) \wedge (\eta_{EP} = 0) \wedge \\ (\eta_A = 0) \wedge (\eta_B = 0) \wedge \\ (\eta_{En} = 1) \\ \top \end{array} \right) \right\}$$

Par exemple, le septième chemin élémentaire $\left(\begin{array}{c} 2.0 \\ \text{slide}^-(SK) \\ B+ \end{array} \right)$ achève la phase S avec l'activation de CycB ($B+$) 2.0 heures après l'inactivation de CycE et CAK ($SK-$). Durant ces deux heures, CycE et CAK sont totalement dégradés ($\text{slide} - (SK)$).

Nous avons alors décrit le graphe d'influence et la trace biologique précédente avec sa postcondition dans un fichier d'entrée pour Holmes BioNet dans le but de simuler l'évolution des entités du graphe au cours du temps.

Tout d'abord, l'analyse de la trace biologique et de la postcondition du cycle cellulaire par Holmes BioNet indique qu'il existe 56 célérités et 130 parties fractionnaires à identifier. Malheureusement, pour cet exemple, l'appel du programme ne nous a pas permis d'obtenir de solution. Le programme construit les contraintes sur les paramètres dynamiques de cette trace biologique, mais le solveur ne trouve pas de solution. Ce problème est dû à une célérité dont le domaine de définition est changeant, et à chaque fois qu'une valeur est fixée pour cette variable dans son domaine de définition, le nouvel appel du solveur ne donne toujours aucune solution à cause sûrement de l'approximation des réels par des flottants. Nous avons alors choisi une valeur "à la main" dans cet intervalle et, en donnant à chaque autre célérité une valeur dans son domaine de définition, nous avons obtenu une solution du CSP et réalisé une simulation. Le résultat est donné en figure 6.5 page ci-contre.

Dans cette figure, nous remarquons que toutes les entités présentent un comportement cyclique sur une période de 22.5 heures. Tous les événements représentés par les atomes de chemin discret sont bien observés dans le graphique et au bon moment (voir figure 9.3 page 153 en annexe pour un aperçu de la simulation entre 20 et 25 heures de la phase mitotique).

Le prototype Holmes BioNet permet d'automatiser le processus d'identification des célérités gouvernant la dynamique de graphes d'influence. A partir de résultats expérimentaux décrits sous forme de traces biologiques et du graphe d'influence du système biologique donné, Holmes BioNet construit les contraintes sur les célérités, cherche à identifier au moins un jeu de paramètres satisfaisant ce CSP, et réalise une simulation du comportement des entités au cours du temps. Ce jeu de paramètres permet donc de comparer les simulations basées sur le paramétrage trouvé avec les données expérimentales.

Les résultats obtenus pour le cycle circadien et pour le cycle cellulaire montrent une concordance entre les traces biologiques d'entrées et les traces simulées. Tous les atomes de chemins discrets sont observés au moment fixé par leurs chemins élémentaires respectifs, et les phénomènes de saturation et dégradation totale des entités biologiques sont également observés.

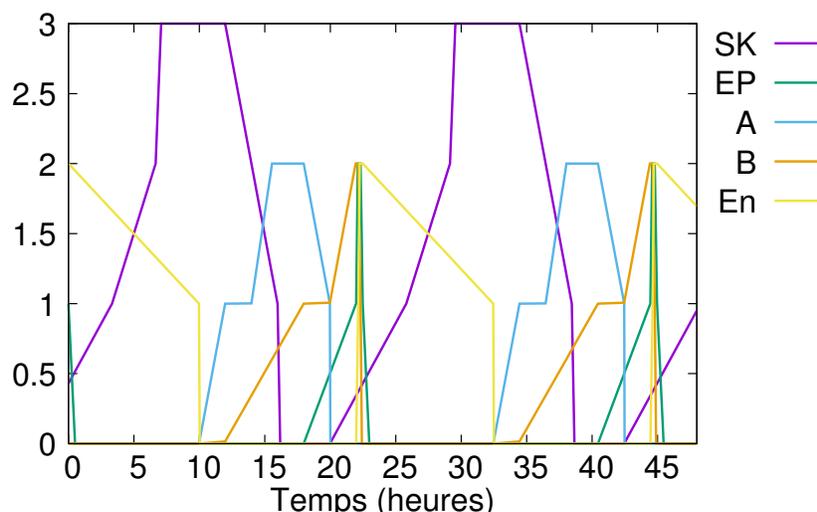


FIGURE 6.5 – Simulation sur 48 heures du cycle cellulaire. Le détail de la phase mitotique (19h - 23h) est donné en annexe figure 9.3 page 153.

Cependant, quelques améliorations restent nécessaires pour permettre l'enchaînement de toutes les étapes du processus. En effet, le solveur de contraintes AbSolute a été conçu pour donner en une seule passe des solutions sûres ou non sûres. Notre approche appelle le solveur plusieurs fois pour identifier un jeu de paramètres qui est solution du CSP. L'appel du solveur mène à approximer des valeurs réelles par des flottants ce qui engendre des contraintes erronées lorsque le solveur est de nouveau appelé.

Néanmoins, Holmes BioNet a montré que le formalisme de la logique de Hoare hybride, couplé à un solveur de contraintes sur les réels, permet d'automatiser la recherche des paramètres d'un modèle hybride. Nous allons voir dans le chapitre suivant comment concevoir une trace biologique du couplage des cycles cellulaire et circadien puis tester Holmes BioNet sur un exemple de plus grande taille.

Chapitre 7

Vers une compréhension du couplage des cycles cellulaire et circadien

Dans les chapitres précédents, nous avons vu qu'il existe un couplage des cycles cellulaires et circadien, et que chacun de ces cycles influence l'autre. A partir des connaissances biologiques préalablement introduites, nous construisons un graphe d'influence avec multiplexes qui représente ce couplage, et nous construisons une trace biologique plausible.

L'enjeu de ce chapitre consiste alors à construire un modèle dynamique du couplage de ces deux cycles en utilisant le prototype Holmes BioNet.

7.1 Graphe d'influence

Le couplage des cycles cellulaire et circadien a été introduit dans la section 2.4 page 40. Ce graphe contient 9 entités (3 pour le cycle circadien et 6 pour le cycle cellulaire). Les influences existantes entre ces deux cycles sont présentées dans le tableau 2.4 page 41. Rappelons que :

- ◇ le complexe protéique PER/CRY active p53 qui inhibe PER2 en retour,
- ◇ REV-ERB α inhibe p21 et il est inhibé par le complexe CycB/Cdk1,
- ◇ BMAL1/CLOCK active la kinase Wee1.

Ces informations montrent qu'il existe des influences du cycle cellulaire sur le cycle circadien, bien qu'elles restent peu étudiées comparées à celles du cycle circadien sur le cycle cellulaire. Réaliser un modèle du couplage à partir des connaissances actuelles pourrait ainsi apporter de nouvelles informations sur les mécanismes existants entre ces deux cycles, et permettrait de comprendre l'adaptation de ces cycles à une perturbation.

Le graphe d'influence avec multiplexes du couplage entre les cycles cellulaire et circadien est détaillé en figure 7.1 page 133. Il reprend les graphes d'influence du cycle circadien présenté

en figure 6.2 page 114, du cycle cellulaire détaillé en figure 6.4 page 126, et inclut les influences entre les deux cycles. Ces influences sont représentées par les multiplexes ayant les formules

$\varphi_{m_{13}}$, $\varphi_{m_{14}}$, $\varphi_{m_{15}}$, $\varphi_{m_{16}}$ et $\varphi_{m_{17}}$:

- ◇ $\varphi_{m_{13}}$ exprime que *p53* inhibe *P* (PER),
- ◇ $\varphi_{m_{14}}$ exprime que *P* active son inhibiteur *p53*,
- ◇ $\varphi_{m_{15}}$ exprime que *BC* (BMAL1/CLOCK) active *En* (la kinase Wee1),
- ◇ $\varphi_{m_{16}}$ exprime que *R* (REV-ERB α) inhibe *En* (*p21*),
- ◇ $\varphi_{m_{17}}$ exprime que *B* (CycB) inhibe *R* (REV-ERB α).

L'action de *p53* sur les deux cycles est aussi intégrée dans le couplage : le mécanisme de surveillance de la transition *G1/S* est intégré à la formule du multiplexe m_{11} . Lorsque *p53* est présent, il permet d'activer *p21* représentée par l'entité *En* ($p53 \rightarrow En$) et bloque ainsi la poursuite du cycle en inhibant les Cdks. Pour que la formule $\varphi_{m_{11}}$ soit active, *p53* doit être présente ou CycE (représentée par l'entité SK) doit être absente pour ne pas inhiber *p27* [116].

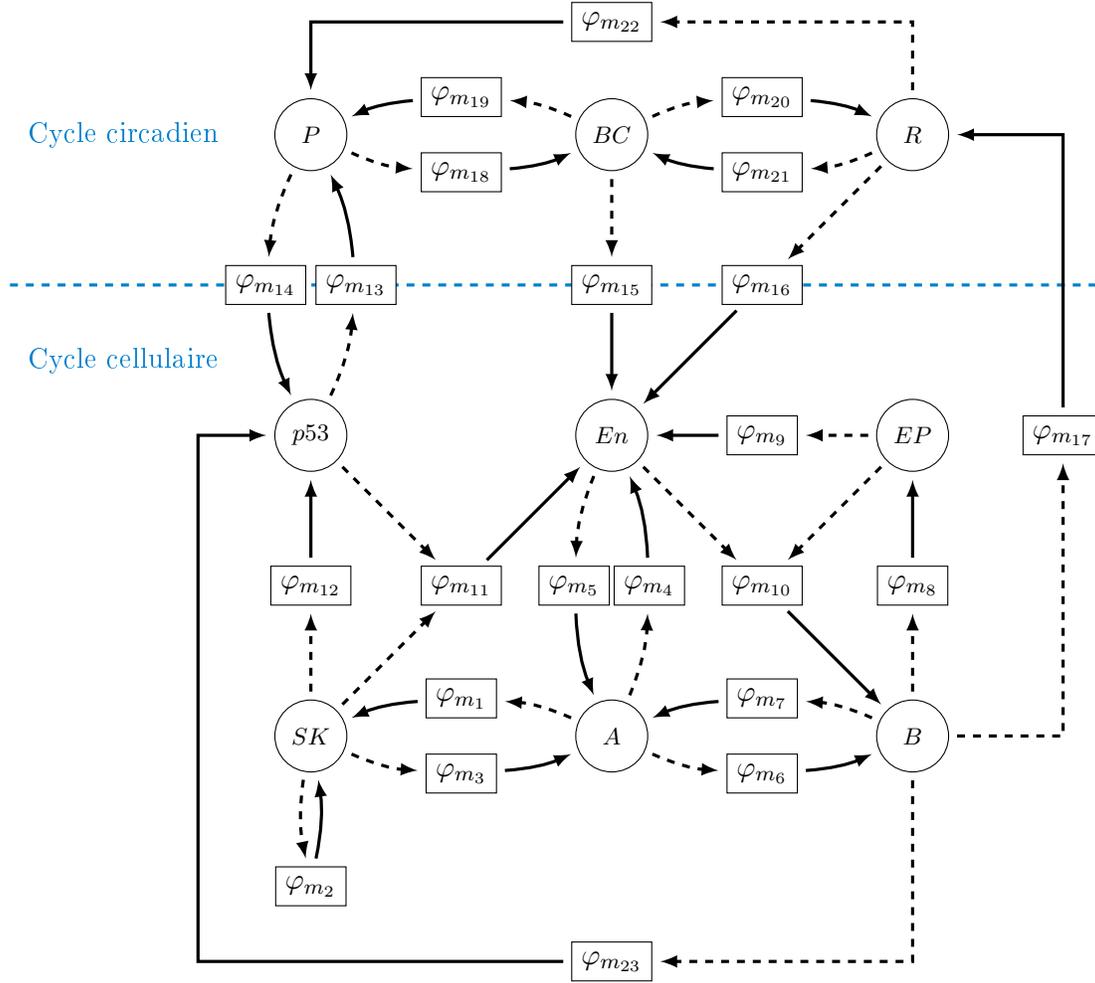
En complément de ce graphe d'influence, nous avons besoin d'identifier une trace biologique plausible pour ce couplage. L'idée est de partir des traces spécifiques de chacun des deux cycles présentées au chapitre précédent et de les combiner pour proposer une trace plausible du couplage tout en restant compatible avec les données expérimentales.

7.2 Verrouillage des phases des deux cycles

Pour pouvoir représenter une trace du couplage, il faut identifier des **points d'ancrage** ou **points temporels** permettant de verrouiller les phases des deux cycles. Ce verrouillage est nécessaire pour relier les traces biologiques spécifiques à chaque cycle et ainsi modéliser le couplage. Ainsi, c'est surtout l'ordre des événements biologiques du couplage combinant les traces biologiques des deux cycles qu'il faut maintenant identifier.

Le premier problème rencontré provient de la différence de période des deux cycles (24 heures pour le cycle circadien et 22.5 heures pour le cycle cellulaire). Puisque l'horloge circadienne synchronise les autres systèmes sur 24 heures, nous avons supposé que le cycle cellulaire présente une période de 24 heures pour verrouiller les phases de ces cycles. En faisant une règle de proportionnalité sur les durées des différentes phases du modèle du chapitre précédent, les nouvelles durées des phases deviennent 10.67 heures pour *G1*, 8.53 heures pour *S*, 4.27 heures pour *G2* et 0.53 heures pour la phase mitotique *M*.

La figure 7.2 page 134 illustre un verrouillage plausible des deux cycles. Le cercle externe représente les événements biologiques qui ont lieu au cours des différentes phases du cycle cellulaire. La phase *G1* démarre lorsque les inhibiteurs de la mitose deviennent inactifs (*dpa EP-*) et termine lorsque les inhibiteurs des Cdks sont inactifs (*dpa En-*). La phase *S* démarre



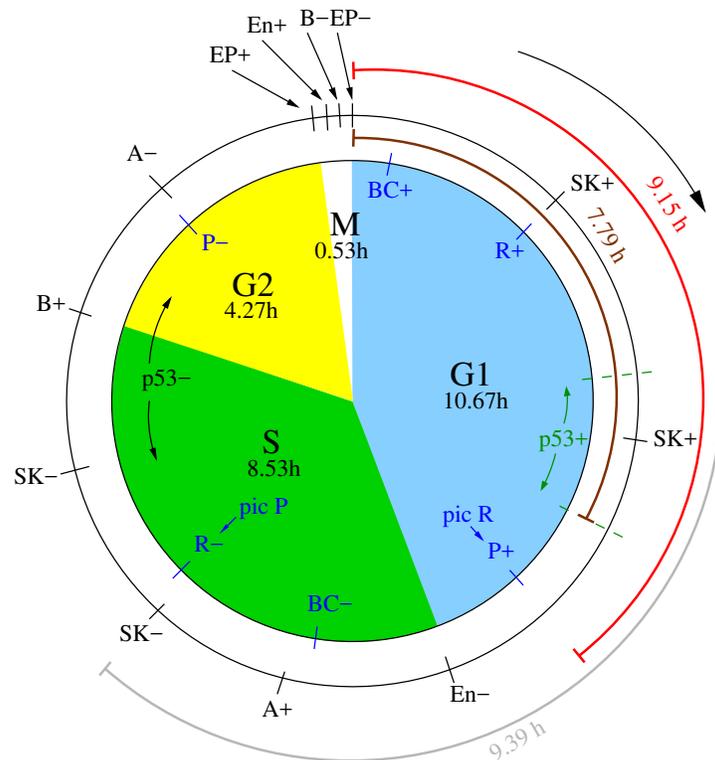
Légendes :

$$\begin{array}{lll}
 \varphi_{m_{11}} = (p53 \geq 1) \vee \neg(SK \geq 1) & \varphi_{m_{14}} = P \geq 1 & \varphi_{m_{17}} = \neg(B \geq 1) \\
 \varphi_{m_{12}} = SK \geq 1 & \varphi_{m_{15}} = BC \geq 1 & \varphi_{m_{23}} = B \geq 1 \\
 \varphi_{m_{13}} = \neg(p53 \geq 1) & \varphi_{m_{16}} = \neg(R \geq 1) &
 \end{array}$$

FIGURE 7.1 – Graphe d'influence du couplage des cycles cellulaire et circadien. Les formules des multiplexes de φ_{m_1} à $\varphi_{m_{10}}$ sont définies dans la figure 6.4 page 126 et celles de $\varphi_{m_{18}}$ à $\varphi_{m_{22}}$ sont définies dans la figure 6.2 page 114.

alors et termine quand la Cycline B est active ($dpa B+$). La phase G2 commence alors et elle termine quand les protéines inhibitrices de la phase M s'activent ($dpa EP+$). Enfin la phase de mitose démarre à $EP+$ et elle termine ($dpa EP-$).

Ensuite, le cercle interne décrit les événements du cycle circadien. Le positionnement de ces événements par rapport à ceux du cycle cellulaire dépend des trois contraintes représentées par les traits rouge, gris et marron. Ces contraintes correspondent à 3 points temporels dans



Légendes :

- REV-ERB α est saturée 9.15 heures après la mitose
- p53 est saturée 7.79 heures après la mitose
- p53 est saturée 9.39 heures avant le pic de PER2

FIGURE 7.2 – Verrouillage de phase entre les cycles cellulaire et circadien. Le cercle interne représente la succession des événements du cycle circadien et le cercle externe la succession des événements du cycle cellulaire.

différents types cellulaire chez le mammifère :

- ◇ la protéine REV-ERB α est saturée 9.15 heures après la mitose (trait rouge) [47],
- ◇ p53 est saturée 7.79 heures après la mitose (trait marron) [24],
- ◇ p53 est saturée 9.39 heures avant le pic de PER2 (trait gris) [58].

Cependant, ces résultats mènent à positionner *p53* sur un créneau de 2 heures, avant ou après l'activation de la Cycline E (second événement *SK+*). Ces éléments ne nous permettent donc pas de verrouiller les phases du cycle circadien et du cycle cellulaire. Nous avons alors choisi de réaliser nos propres expériences pour valider ou réfuter les précédents résultats mais également pour disposer d'un jeu de données complet où on a une mesure expérimentale pour chaque entité du modèle dans un même type cellulaire.

7.3 Expériences

Nous avons réalisé les expériences sur une lignée cellulaire immortalisée nommée HaCat qui sont des cellules de la peau appelées kératinocytes chez l’homme adulte qui présente une horloge et un cycle cellulaire synchronisable.

Dans un premier temps, nous avons synchronisé les cellules par déprivation de sérum (arrêt des cellules en $G0 - G1$) suivi d’une restimulation. Les cellules vont ensuite redémarrer leur progression en début de phase $G1$. Nous vérifions alors que les cellules sont bien synchronisées, c’est-à-dire que les cellules parcourent les différentes phases du cycle cellulaire en même temps. L’outil permettant cette vérification est la **cytométrie de flux FACS** (*fluorescence-activated cell sorting*) qui analyse la taille et le contenu en ADN des cellules une par une en les passant dans un faisceau laser ce qui permet de les distinguer selon leurs caractéristiques (les phases du cycle cellulaire dans notre cas).

La figure 7.3 page suivante montre la proportion de cellules dans les différentes phases du cycle cellulaire sur un intervalle de temps de 24 heures. Nous analysons les résultats à partir de 16 heures car il existe un temps de latence nécessaire pour que les cellules à l’état physiologique de repos transitent de la phase $G0$ vers la phase $G1$.

Les résultats obtenus montrent que les cellules sont synchronisées car 70%¹ des cellules sont en phase $G1$ à $t = 16h$, puis le pourcentage de cellules en phase S augmente 8 à 10 heures plus tard (environ 40% car des cellules restent en $G1$ plus ou moins longtemps). La phase $G2$ puis la phase mitotique démarrent autour de $t = 32h$ et permet de démarrer un nouveau cycle cellulaire à $t = 36h$. Les cellules sont donc synchronisées sur une période de 20 heures environ.

Puisque les cellules sont bien synchronisées, nous vérifions que les gènes de l’horloge ont aussi une activité rythmique en analysant quantitativement leurs ARNs (*rev-erba*, *bmali*) toutes les 4 heures pendant 24 heures, voir figure 7.4 page suivante. Ces résultats montrent que les gènes de l’horloge oscillent sur la période d’environ 20 à 22 heures, et que le pic d’ARN de *rev-erba* est observé durant la phase $G1$ (à 18 heures). Ce résultat indique que le pic de protéine de REV-ERB α s’observe plus tard, à la fin de la phase $G1$ comme observé dans des travaux précédents [47].

Les expériences précédentes ont permis de montrer la synchronisation des cellules, et l’oscillation des gènes de l’horloge circadienne. Dans un premier temps, pour observer le comportement des protéines du cycle cellulaire et de l’horloge, nous avons souhaité analyser par FACS les protéines présentes avec une analyse toutes les 2 heures pendant 36 heures. Les protéines sont détectées par des **anticorps**² couplés à des composés fluorescents appelés fluorochromes.

1. La synchronisation des cellules n’est jamais totale, quelque soit la technique employée.

2. Dans notre cas, un anticorps se définit comme une protéine qui réagit en se fixant sur son composé cible.

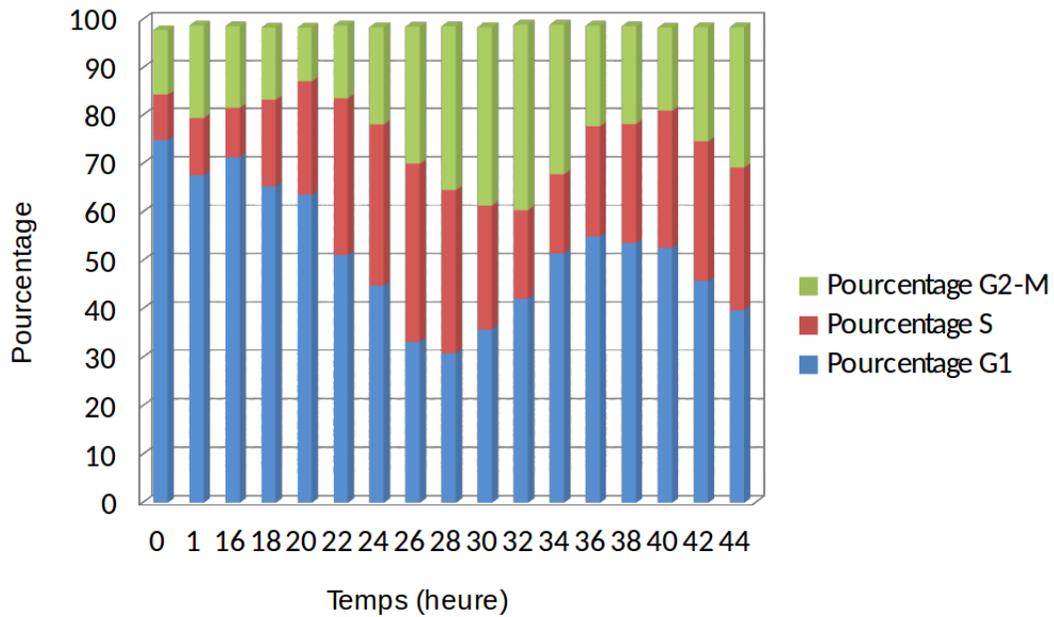


FIGURE 7.3 – Synchronisation des cellules HaCat. Cet histogramme décrit la proportion des cellules dans les différentes phases du cycle cellulaire.

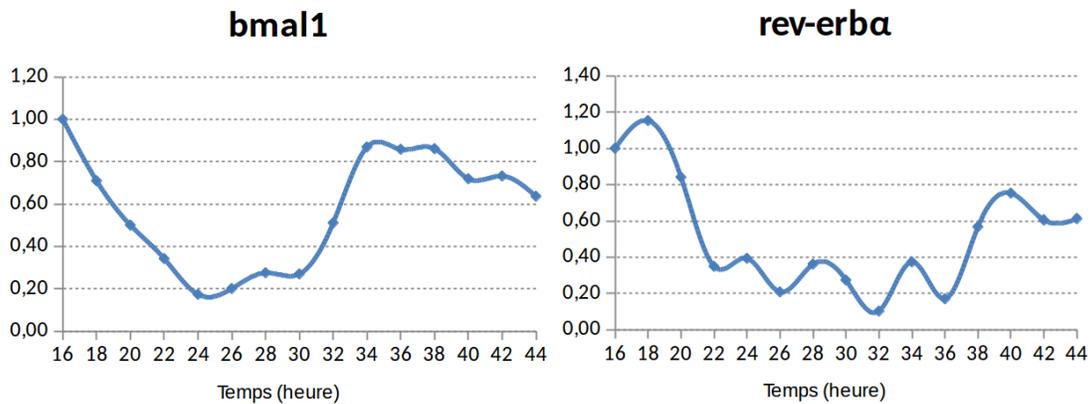


FIGURE 7.4 – Evolution de la quantité d'ARNs issus des gènes *bmal1* et *rev-erba* dans les cellules HaCat synchronisées.

La fixation des anticorps sur leurs protéines cibles permet aux fluorochromes d'émettre de la lumière à une longueur d'onde spécifique lorsqu'ils traversent le faisceau du laser du cytomètre de flux. Les cellules seront donc triées selon la longueur d'onde qu'elles émettent et, *a fortiori*, selon la phase du cycle cellulaire. La méthode FACS a l'avantage de faire plusieurs analyses de différents critères en même temps et qu'il est possible d'analyser des milliers de cellules de manière unique et très rapidement.

Les résultats obtenus ne nous ont pas permis d'observer des comportements oscillants pour les protéines du cycle cellulaire et les protéines de l'horloge circadienne. La raison de tels résultats est que la sensibilité d'une telle approche s'avère insuffisante. En effet, seule la Cycline B a montré un comportement cyclique. Le problème pourrait alors venir de la perméabilisation des cellules au niveau de la membrane plasmique ou nucléaire.

Avant de se concentrer sur l'activité protéique des différents composés intervenant dans les cycles circadien et cellulaire avec les techniques de Western Blot classiques, nous avons décidé de nous intéresser d'abord au profils de leurs ARNs. Puisque les réseaux de régulation étudiés présentent des boucles, l'accumulation des ARNs reflète l'activité des protéines du réseau. Nous avons analysé les ARNs des gènes de l'horloge (*bmal1*, *per*, *rev-erb α*) et du cycle cellulaire (*cycline E*, *cycline A*, *cycline B*, *p53*, *p27*, *wee1*), voir figure 7.5 page suivante. L'identification de la quantité en ARN avec la technique de PCR quantitative en temps réel dans des cellules HaCat synchronisées provient de deux expériences : la première consiste à bloquer les cellules en phase *G2* avec le composé nocodazole et la seconde bloque les cellules au début de la phase *S* avec l'ajout du composé thymidine. Ce blocage a lieu pendant plusieurs heures le temps de synchroniser toutes les cellules puis le composé qui bloque les cellules est retiré. La quantité en ARN est ensuite identifiée. Ces deux expériences sont réunies pour reconstituer un cycle de la population hautement synchronisé.

L'ensemble des résultats que nous obtenons est cohérent avec les expériences de séquençage d'immunoprécipitation de chromatine (ChIP) pour les gènes de l'horloge présentés dans la sous-section 6.5.1 page 123. Les résultats obtenus pour les gènes du cycle cellulaire sont également en accord avec les connaissances biologiques actuelles qui sont représentées dans la trace biologique du cycle cellulaire de la sous-section 6.5.2 page 125. Cependant, nous pouvons tout de même remarquer que l'ARN du gène *cycline B* est présent durant la phase *G1*. Ce résultat pourrait néanmoins correspondre à un problème de mesure puisque l'activité de la cycline B s'arrête à partir de la phase mitotique, ou encore que la protéine Cycline B est plus régulée que son ARN. Une nouvelle expérience ou une meilleure résolution de l'expérience nous permettra de décider de la validité ou non de ce pic en ARN durant la phase *G1*.

Malheureusement, ces résultats restent insuffisants pour verrouiller les phases du cycle cellulaire et du cycle circadien et permettre de représenter une trace biologique du couplage car la

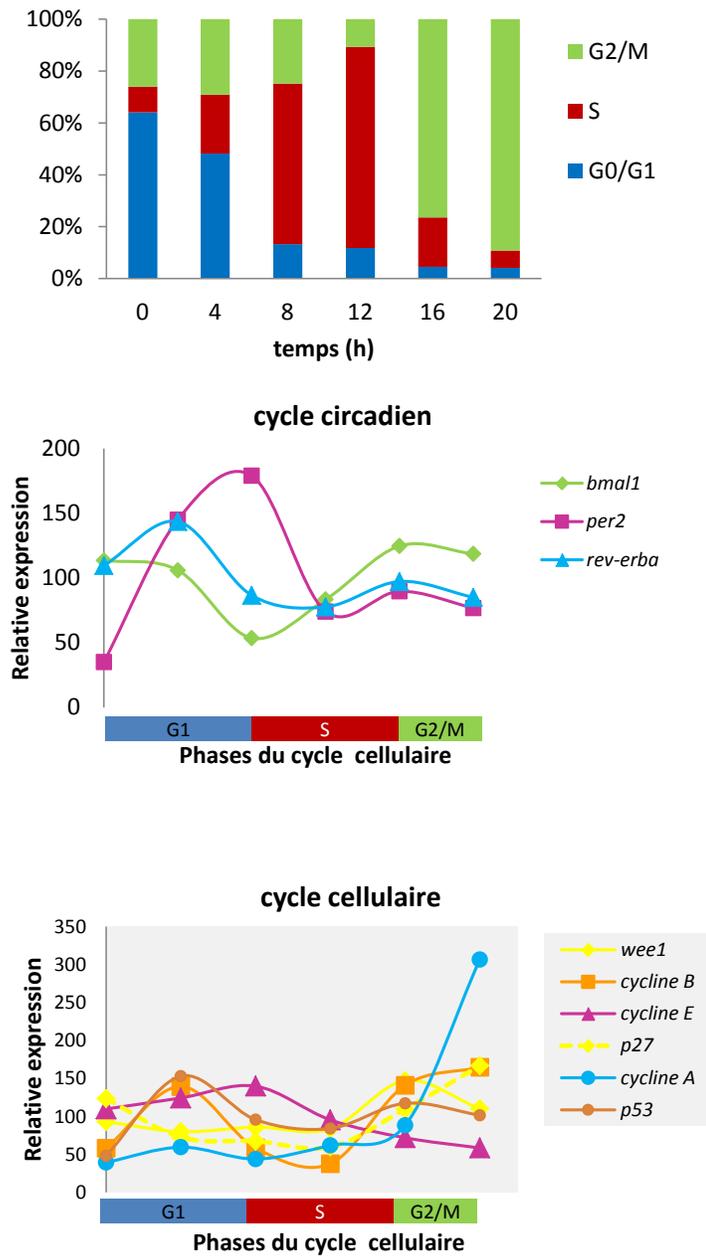


FIGURE 7.5 – Evolution des ARNs sur 20 heures. L’histogramme décrit le pourcentage de cellules dans les différentes phases du cycle cellulaire. Les deux figures suivantes représentent respectivement l’expression en ARN des gènes du cycle circadien puis du cycle cellulaire.

résolution (4 heures) n'est suffisamment pas précise. Nous proposons alors une trace biologique théorique basée sur les données connues et sur des hypothèses permettant de verrouiller les phases des deux cycles.

7.4 Construction du triplet de Hoare

La trace biologique et la postcondition que nous construisons s'appuient sur les résultats précédents et les connaissances biologiques présentées dans l'ensemble des chapitres précédents. Les points temporels présentés en section 7.2 page 132 permettent de coordonner les deux cycles sans trop de précision. Dans nos résultats, le pic d'ARNs issu du gène *rev-erba* en phase *G1* est en accord avec le pic de sa protéine qui a lieu 9.15 heures après la mitose (*dpa P+*) présenté en [47]. De plus, nous avons choisi le point temporel pour lequel *p53* est saturée 9.39 heures avant le pic de *PER2* (*dpa R-*) (et donc *p53* est actif au moins 9.39 heures avant). De plus, nous avons fait le choix de positionner le *dpa p53-* avant le *dpa B+* puisque la Cycline B active *p53*. Nous construisons alors la trace biologique suivante :

$$\begin{aligned}
 & \left(\begin{array}{c} 0.59 \\ \top \\ BC+ \end{array} \right); \left(\begin{array}{c} 2.44 \\ \text{slide}^+(BC) \wedge \text{slide}^-(P) \\ R+ \end{array} \right); \left(\begin{array}{c} 0.52 \\ \text{slide}^-(EP) \\ SK+ \end{array} \right); \left(\begin{array}{c} 1.77 \\ \top \\ p53+ \end{array} \right); \left(\begin{array}{c} 1.78 \\ \top \\ SK+ \end{array} \right); \left(\begin{array}{c} 2.05 \\ \text{slide}^+(R) \\ P+ \end{array} \right); \\
 & \left(\begin{array}{c} 1.51 \\ \text{slide}^+(SK) \\ En- \end{array} \right); \left(\begin{array}{c} 1.93 \\ \top \\ BC- \end{array} \right); \left(\begin{array}{c} 0.20 \\ \text{slide}^-(En) \\ A+ \end{array} \right); \left(\begin{array}{c} 2.13 \\ \top \\ SK- \end{array} \right); \left(\begin{array}{c} 0.11 \\ (\text{slide}^-(BC) \wedge \\ \text{slide}^+(P)) \\ R- \end{array} \right); \left(\begin{array}{c} 2.02 \\ \text{slide}^+(A) \\ SK- \end{array} \right); \\
 & \left(\begin{array}{c} 1.06 \\ \top \\ p53- \end{array} \right); \left(\begin{array}{c} 1.07 \\ \text{slide}^-(SK) \\ B+ \end{array} \right); \left(\begin{array}{c} 1.97 \\ \top \\ P- \end{array} \right); \left(\begin{array}{c} 0.16 \\ \top \\ A- \end{array} \right); \left(\begin{array}{c} 2.14 \\ \text{slide}^+(B) \\ EP+ \end{array} \right); \left(\begin{array}{c} 0.18 \\ \text{slide}^+(EP) \\ En+ \end{array} \right); \\
 & \left(\begin{array}{c} 0.18 \\ \text{slide}^-(A) \wedge \text{slide}^+(En) \\ B- \end{array} \right); \left(\begin{array}{c} 0.19 \\ \text{slide}^-(B) \\ EP- \end{array} \right) \left\{ \left(\begin{array}{c} (\eta_P = 0) \wedge (\eta_{BC} = 0) \wedge \\ (\eta_R = 0) \wedge (\eta_{SK} = 0) \wedge \\ (\eta_{EP} = 0) \wedge (\eta_A = 0) \wedge \\ (\eta_B = 0) \wedge (\eta_{En} = 1) \wedge \\ (\eta_{PP} = 0) \\ \top \end{array} \right) \right\}
 \end{aligned}$$

Cette trace combine les traces biologiques du cycle circadien (en rouge) et du cycle cellulaire décrites dans le chapitre précédent. Nous créons alors le fichier d'entrée nécessaire à Holmes BioNet pour identifier un jeu de paramètres compatible avec cette trace plausible.

7.5 Utilisation du prototype et simulation

Tout d'abord, Holmes BioNet construit les contraintes sur les célérités du modèle pour que ce dernier soit compatible avec la trace biologique. Le programme identifie 1536 contraintes qui dépendent de 530 variables (152 célérités et 378 parties fractionnaires) parmi lesquelles 35 parties fractionnaires de sortie sont des constantes (20 issues du *dpa* et 15 des assertions *slide*).

Le premier appel du solveur identifie uniquement des solutions non sûres et le temps de calcul pour obtenir ces résultats est de 10 minutes environ.

Similairement à l'étude du cycle cellulaire, Holmes BioNet ne donne pas de solution pour la trace biologique du couplage. En effet, le premier appel du solveur retourne une solution non sûre. Une valeur est alors affectée à une variable dans son domaine de définition changeant puis intégrée au CSP courant, et le solveur analyse le nouveau CSP. Ce procédé est réitéré et ne permet pas de trouver de solution. Nous avons alors choisi un jeu de paramètres "à la main" après le premier appel du solveur pour les variables dont le domaine est changeant. Cependant, ce jeu de paramètres n'est pas solution du CSP mais permet de réaliser une première simulation pour comprendre l'évolution des différentes entités au cours du temps, voir figure 7.6 page ci-contre. On pourrait utiliser de telles simulations pour proposer d'éventuelles nouvelles expériences afin de compléter la connaissance sur le système biologique, qui pourront renforcer les contraintes du CSP et faciliter sa résolution.

7.6 Discussion

Tout d'abord, nous pouvons remarquer que, après 35 heures, le modèle atteint un état stable où les niveaux d'expression hybrides de *SK*, *EP*, *B* et *BC* valent 2.0, et les niveaux d'expression hybrides des autres entités valent 0.0. Ce comportement n'est pas conforme à la trace biologique donnée.

Le graphique montre une cohérence des changements des niveaux d'expression des entités lors des 6 premiers chemins élémentaires de la trace biologique. Ensuite, le septième chemin élémentaire où le *dpa* est *En-* n'est pas observé car c'est le niveau d'expression hybride de l'entité *BC* qui diminue à 16.23 heures (*dpa BC-*) avant celui de l'entité *En* à 16.3 heures. Ensuite le niveau d'expression hybride de *A* augmente à 18.43 heures comme attendu, suivi par une diminution de *R* (18.67 heures) puis de *SK* deux fois (20.56 heures et 22.69 heures). Les évé-

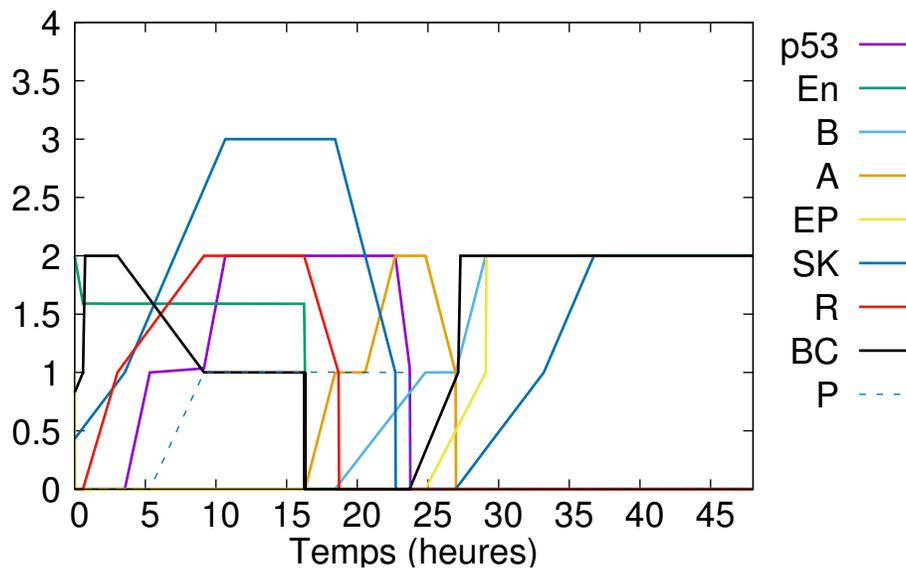


FIGURE 7.6 – Simulation sur 48 heures du couplage des cycles circadien et cellulaire.

nements suivants sont la diminution du niveau d'expression hybride de P (23.70 heures) et $p53$ (23.72 heures) puis l'augmentation du niveau d'expression hybride de B à 24.82 heures. Les saturations ou dégradations totales des entités données dans la trace biologique sont aussi observées durant les événements précédemment cités.

Bien que ces événements n'aient pas tous lieu dans l'ordre donné par la trace biologique, ces événements permettent d'atteindre un état que le modèle doit atteindre dans la trace biologique donnée. Cependant, l'événement suivant est l'augmentation du niveau d'expression hybride de BC à 27.14 heures qui est attendu 3.44 heures plus tard. De manière plus surprenante, l'entité En n'augmente pas, bien que son activateur BC est actif. La cycline B (entité B) n'est plus dégradée et la division de la cellule n'a pas lieu, menant à un état stable.

Ces résultats mènent aux conclusions suivantes :

- ◊ La trace exhibée présente des événements inversés par rapport à la trace biologique plausible donnée en argument. Ce résultat peut s'expliquer car les données de l'horloge circadienne et du cycle cellulaire proviennent de mesures expérimentales différentes, d'un côté l'analyse de la quantité des protéines, et de l'autre leurs activités. Toutes les activités des protéines de l'horloge sont alors décalées par rapport aux activités de leurs gènes. De plus, les mesures ne sont pas issues du même type cellulaire puisque les données du cycle circadien proviennent du foie chez la souris tandis que les données du cycle cellulaire

proviennent de la peau chez l'homme. Par ailleurs, nous ne pouvons pas exclure que les relations de phase entre les différentes protéines de l'horloge soient identiques dans le foie des souris et dans les cellules en prolifération chez l'homme.

- ◇ L'entité En ne semble plus augmenter malgré son inhibiteur A inactif et ses activateurs EP et BC actifs. Nous pensons que le jeu de paramètres choisi ne contraint pas suffisamment les célérités de En et les valeurs fixées arbitrairement à ces célérités ne permettent pas d'augmenter le niveau d'expression de En quand toutes les conditions le permettent.

Ces résultats restent toutefois intéressants pour les biologistes et peuvent mener à de nouvelles expériences. En effet, on peut envisager de vérifier l'ordre des activités des protéines des cycles circadien et cellulaire plus précisément que les expériences proposées dans la section précédente. Par exemple, des expériences pourraient confirmer ou infirmer que le complexe BMAL1/CLOCK diminue ($BC-$) avant les protéines inhibitrices des phases $G1$ et $G2$ ($En-$).

Nous pensons qu'il est possible d'obtenir une trace exhibée du couplage en accord avec la trace biologique lorsque l'ordre des événements sera correctement identifié. En effet, les résultats du cycle circadien et du cycle cellulaire seuls ont démontré la cohérence entre les données d'entrée et la dynamique du modèle obtenu. Ces résultats restent prometteurs pour modéliser des systèmes biologiques à partir d'observations expérimentales.

Chapitre 8

Conclusion

La modélisation de réseaux de régulation est une aide précieuse pour l'étude des systèmes biologiques complexes et devient une partie intégrante de leur analyse aux côtés des approches expérimentales. C'est cette complexité qui permet au système biologique de maintenir les homéostasies nécessaires et l'adaptation aux perturbations de l'environnement. L'activité de modélisation de la dynamique de ces systèmes permet non seulement de questionner les frontières du sous-système étudié (a-t-on oublié une entité régulatrice majeure ? les abstractions faites sont-elles trop drastiques ?) mais surtout de comprendre davantage le système biologique étudié. Il est cependant indispensable d'émettre des hypothèses, et en les confrontant aux modèles précédemment construits, il devient possible de les réfuter en montrant leurs incohérences avec les conséquences des connaissances actuelles.

Les méthodes formelles ont été adaptées aux systèmes biologiques pour identifier les états accessibles du système (analyse statique), pour vérifier des propriétés de logiques temporelles (model checking), ou encore pour vérifier des systèmes par vérification déductive. Ces méthodes de vérification peuvent être adaptées aux modèles de systèmes biologiques permettant à la communauté scientifique de vérifier que ces modèles satisfont un certain nombre de spécifications (atteignabilité, formules temporelles, cohérence par rapport à une trace).

8.0.1 Contributions

La modélisation de la dynamique des réseaux de régulation biologique présuppose un cadre de modélisation pré-établi, qui doit être choisi en fonction non seulement des connaissances disponibles sur le système biologique étudié mais aussi du niveau d'abstraction souhaité. Le cadre de modélisation hybride sur lequel s'appuie nos travaux présenté dans cette thèse, présuppose une description des systèmes biologiques par un graphe d'influence avec multiplexes, souvent de petite taille, et la dynamique associée repose sur une sorte d'automate hybride de plus grande

taille : la dynamique du modèle est somme toute une trajectoire linéaire par morceau qui doit rester cohérente avec l'ensemble des observations. Ces trajectoires linéaires par morceaux présentent l'avantage de permettre une recherche des paramètres dynamiques qui peut être assistée par ordinateur.

Le cadre de modélisation hybride s'adapte remarquablement aux méthodes formelles. Nous avons proposé une logique de Hoare hybride dont le calcul de la plus faible précondition permet de construire un ensemble de contraintes minimal que doivent satisfaire les paramètres dynamiques du modèle pour le rendre compatible avec une trace biologique observée. A partir du graphe d'influence avec multiplexes et d'observations expérimentales représentées comme une succession d'événements espacés par des durées, le calcul de la plus faible précondition construit pas à pas ces contraintes minimales. Notons que d'un point de vue expérimental, la dégradation totale et/ou la saturation totale de certaines protéines sont observables. De telles informations sont pertinentes pour la modélisation et sont intégrées au formalisme de description des traces observées. Cette logique de Hoare en temps continu a été montrée correcte et complète sémantiquement (le calcul de la plus faible précondition calcule bien la plus faible précondition).

La démonstration de la faisabilité a pris pour exemple la modélisation du cycle circadien, du cycle cellulaire puis du couplage de ces deux cycles. Le calcul de la plus faible précondition de la logique de Hoare hybride a permis de construire de façon automatisée les contraintes linéaires (lorsque tous les temps de séjour dans les états sont connus) sur les paramètres dynamiques associés à chaque état parcouru par les traces biologiques données. La taille de ces problèmes de satisfaction de contraintes (CSP) varie en fonction du nombre d'entités, du nombre de régulations dans le graphe d'influence, mais aussi du nombre d'événements présents dans la trace biologique.

Pour permettre de simuler les modèles, il est alors nécessaire d'instancier les paramètres dynamiques du modèle, et cela mène à vouloir trouver des solutions au problème de satisfaction de contraintes (CSP) construit précédemment. Malheureusement, la taille de ces CSP est trop grande pour qu'ils puissent être résolus "à la main". Nous avons cependant montré qu'il est possible de les résoudre en utilisant des solveurs de contraintes sur les réels. Le solveur AbSolute identifie un jeu de paramètres satisfaisant les CSPs associés aux graphes d'influence, et ce jeu de paramètres reste cohérent avec la trace biologique spécifiée. Le temps de résolution du solveur semble être exponentiel en fonction du nombre de variables présentes dans le graphe d'influence.

Le programme Holmes BioNet implémente les idées présentées dans cette thèse. Ce programme regroupe les trois éléments nécessaires : le calcul de la plus faible précondition de la logique de Hoare hybride (pour la construction du CSP), une stratégie d'appels au solveur de contraintes AbSolute (pour essayer d'obtenir une ou plusieurs solutions du CSP), et un simu-

lateur qui permet de simuler la dynamique des entités du graphe d'influence lorsqu'un jeu de paramètres est identifié. Toutefois, quelques améliorations du programme sont à envisager. En effet, la stratégie d'appels au solveur AbSolute peut ne retourner aucune solution alors qu'une solution existe ; c'est ce qui s'est passé pour la modélisation du cycle cellulaire.

8.0.2 Perspectives

La première difficulté que rencontre l'utilisateur de la démarche proposée dans cette thèse est l'écriture de la trace biologique expérimentale ou hypothétique. La syntaxe peut être un peu difficile à appréhender, mais le plus délicat est de ne pas écrire dans cette trace des informations qui vont se révéler incompatibles : par exemple, écrire pour deux chemins élémentaires différents que les célérités d'une même variable sont de signe contraire alors que les deux célérités doivent être de même signe puisque soumises aux mêmes ressources. Si de telles informations mènent à des contraintes opposées, le problème de satisfaction de contraintes n'est pas solvable. Pour pallier ce problème, une aide à la modification de la trace biologique pourrait être proposée en indiquant à l'utilisateur les informations incompatibles de la trace biologique originelle.

Le calcul de la plus faible précondition se base essentiellement sur la capacité d'écrire une relation analytique entre le point d'entrée et le point de sortie dans un même état qualitatif. La démarche serait donc généralisable à tout système hybride où une telle relation analytique est possible. La première extension possible de cette logique de Hoare hybride, munie de son calcul de plus faible précondition, pourrait donc se focaliser sur les modèles de réseaux d'influence par systèmes d'équations différentielles linéaires par morceau (Piecewise Linear Differential Equation ou PLDE) : les relations analytiques entre points d'entrée et de sortie des différents états qualitatifs sont simplement déduites de l'expression analytique des solutions du système différentiel dans ces états qualitatifs. A l'instar de la logique de Hoare hybride présentée dans ce document, les discontinuités des modèles PLDE proviennent de la simplification de phénomènes non linéaires. Dans la plupart des cas, les transitions discrètes sont "transparentes" pour maintenir une trajectoire réaliste comparée aux observations.

L'un des grands domaines de la médecine pour laquelle la logique de Hoare hybride pourrait montrer un intérêt est la recherche translationnelle en oncologie. En effet, c'est un domaine où l'approche chronothérapeutique a déjà montré son intérêt.

"Tout médicament d'ailleurs n'est pas également bien indiqué à toute heure, et ici encore la période diurne a besoin d'être consultée. [...] Des médicaments utiles en eux-mêmes, mais intempestifs, peuvent exciter des perturbations fatales, si l'on néglige l'étude des époques diurnes."

Chapitre 8 : Conclusion

Extrait de *Éphémérides de la vie humaine, ou Recherches sur la révolution journalière et la périodicité de ses phénomènes dans la santé et les maladies*, Thèse du médecin français Julien-Joseph Virey, 1814

Cette citation présente les premières observations sur l'importance d'administrer des médicaments à des heures où l'organisme répond le mieux. La **chronothérapie**, domaine cherchant à identifier les heures optimales de la prise de médicament, a pour but de répondre à ce besoin. Derrière cette idée d'optimisation de traitement, il y a la volonté d'améliorer son effet, de diminuer les effets indésirables et la dose administrée.

Actuellement, la chronothérapie est peu répandue dans les traitements de tous les jours et dans les traitements à effets secondaires importants [115]. Pourtant, des études ont montré une meilleure efficacité de candidats anticancéreux (qui ont été abandonnés durant les essais cliniques à cause de résultats insatisfaisants) lorsqu'ils sont administrés de façon chronomodulée, comparé à des traitements actuellement sur le marché. C'est notamment le cas de l'oxaliplatine combiné avec le fluorouracil et l'acide folinique [87] où la toxicité de ce traitement a été réduite par 2 et son effet a été amélioré de 76% simplement en choisissant les bonnes heures d'administration.

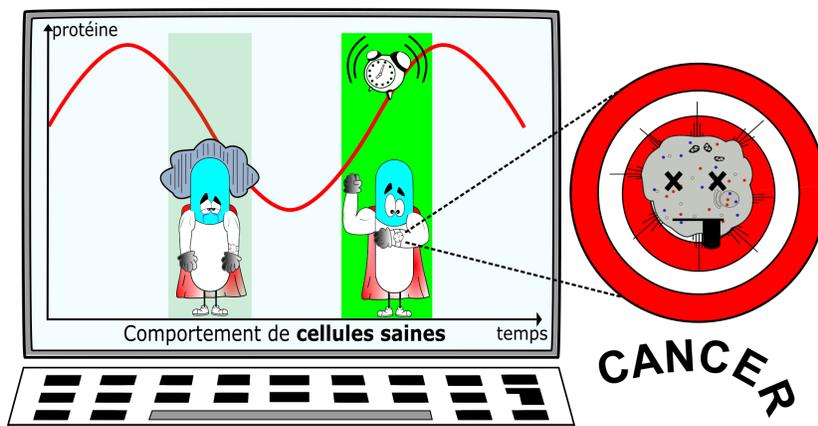


FIGURE 8.1 - "A l'heure des super médicaments". Schéma illustrant la prise de médicament optimale par des outils informatiques. Les médicaments agissent différemment selon l'intervalle de temps où ils sont administrés. A la différence du médicament de gauche, le médicament de droite ne perturbe pas les cellules saines et cible alors spécifiquement les cellules cancéreuses. Ce schéma a été utilisé pour présenter ces travaux de thèse durant le concours "Ma thèse en 180 secondes" [13].

En cancérologie, la chronothérapie permettrait de cibler spécifiquement les cellules cancéreuses désynchronisées lorsque les cellules saines synchronisées ne sont pas ou peu perturbées

par le traitement.

Lorsque le modèle du couplage des cycles cellulaire et circadien de cellules saines sera en accord avec les traces biologiques données, et en connaissant l'effet du traitement sur le couplage, nous envisageons d'étudier l'effet de l'heure d'administration du médicament sur le modèle en l'administrant virtuellement dans ce modèle à différents intervalles de temps. Quand le traitement ne montre pas de perturbation de la dynamique du modèle des cellules saines, l'intervalle de temps où ce traitement est administré est un créneau potentiel ciblant spécifiquement les cellules cancéreuses désynchronisées, voir figure 8.1 page précédente. Pour s'assurer de la validité de cette démarche, il faudra d'abord envisager de tester des traitements anticancéreux pour lesquels une administration chronomodulée a déjà montré son efficacité par rapport à une administration classique, et voir si l'étude de l'effet de l'heure d'administration du médicament sur le modèle fait ressortir les mêmes plages horaires d'administration que l'approche empirique *in vivo*.

Une première piste proposée consiste à automatiser la recherche de l'heure idéale pour administrer des traitements anticancéreux en couplant les modèles présentés ici avec d'autres modèles liés à des phénotypes particuliers. Cette analyse permettrait d'accompagner le biologiste dans la recherche de l'heure optimale d'administration et dans la recherche de nouveau traitement anticancéreux. Ce type d'application aura pour conséquence de commencer à prendre en compte une partie du comportement global de l'organisme (cycles circadien et cellulaire), de prendre en compte des particularités de l'individu (médecine personnalisée) dans le but d'augmenter les effets bénéfiques tout en diminuant les doses et les effets indésirables.

Chapitre 9

Annexes

9.1 Contraintes du rétrocontrôle négatif

Après simplification et réduction des contraintes construites sur l'exemple de rétrocontrôle négatif donné en section 4.6, en considérant une trace biologique cyclique (voir section 5.4, page 106) et en incluant les contraintes entre célérités voisines (second axiome en sous-section 4.3.5, page 82), nous obtenons 6 valeurs exactes sur des parties fractionnaires et 32 contraintes :

$$\begin{aligned} & (\pi_{v_1}^{3'} = 0.) \wedge (\pi_{v_2}^{3'} = 0.) \wedge (\pi_{v_2}^{2'} = 0.) \wedge (\pi_{v_1}^{1'} = 1.) \wedge \\ & (\pi_{v_2}^{1'} = 1.) \wedge (\pi_{v_2}^{0'} = 1.) \wedge (\pi_{v_1}^4 = \pi_{v_1}^0) \wedge (\pi_{v_2}^4 = \pi_{v_2}^0) \wedge \\ & (C_{v_1, \emptyset, 0} \times C_{v_1, \emptyset, 1} \geq 0.) \wedge (C_{v_1, \{m2\}, 0} \times C_{v_1, \{m2\}, 1} \geq 0.) \wedge \\ & (C_{v_2, \emptyset, 0} \times C_{v_2, \emptyset, 1} \geq 0.) \wedge (C_{v_2, \{m1\}, 0} \times C_{v_2, \{m1\}, 1} \geq 0.) \wedge \\ & (\pi_{v_1}^{3'} = 1. - \pi_{v_1}^4) \wedge (\pi_{v_2}^{3'} = \pi_{v_2}^4) \wedge \\ & (\pi_{v_1}^{2'} < 1.) \wedge (\pi_{v_1}^{2'} > 0.) \wedge (\pi_{v_2}^{2'} = 1. - \pi_{v_2}^3) \wedge (\pi_{v_1}^{2'} = \pi_{v_1}^3) \wedge \\ & (\pi_{v_1}^{1'} = 1. - \pi_{v_1}^2) \wedge (\pi_{v_2}^{1'} = \pi_{v_2}^2) \wedge \\ & (\pi_{v_1}^{0'} < 1.) \wedge (\pi_{v_1}^{0'} > 0.) \wedge (\pi_{v_2}^{0'} = 1. - \pi_{v_2}^1) \wedge (\pi_{v_1}^{0'} = \pi_{v_1}^1) \wedge \\ & (\pi_{v_2}^0 = \pi_{v_2}^{0'} - C_{v_2, \{m1\}, 0} \times 5.) \wedge (C_{v_2, \{m1\}, 0} > 0.) \wedge \\ & (\pi_{v_1}^0 = \pi_{v_1}^{0'} - C_{v_1, \emptyset, 0} \times 5.) \wedge (C_{v_2, \{m1\}, 1} > 0.) \wedge \\ & (\pi_{v_1}^1 = \pi_{v_1}^{1'} - C_{v_1, \{m2\}, 0} \times 7.) \wedge (C_{v_1, \{m2\}, 0} > 0.) \wedge \\ & (\pi_{v_2}^2 = \pi_{v_2}^{2'} - C_{v_2, \emptyset, 1} \times 8.) \wedge (C_{v_2, \emptyset, 1} < 0.) \wedge \\ & (\pi_{v_1}^2 = \pi_{v_1}^{2'} - C_{v_1, \{m2\}, 1} \times 8.) \wedge (C_{v_2, \emptyset, 0} < 0.) \wedge \\ & (\pi_{v_1}^3 = \pi_{v_1}^{3'} - C_{v_1, \emptyset, 1} \times 4.) \wedge (C_{v_1, \emptyset, 1} < 0.) \wedge \\ & (\pi_{v_2}^1 > \pi_{v_2}^{1'} - C_{v_2, \{m1\}, 1} \times 7.) \wedge (\pi_{v_2}^3 < \pi_{v_2}^{3'} - C_{v_2, \emptyset, 0} \times 4.); \end{aligned}$$

La résolution du CSP présentant les contraintes décrites ci-dessus mène à une solution sûre. A partir des domaines de définition des parties fractionnaires ($[0, 1]$), les valeurs obtenues des

positions satisfaisant les contraintes sont détaillées dans le tableau 9.1.

$\pi_{v_1}^0 = 1.0$	$\pi_{v_2}^0 = 0.0$
$\pi_{v_1}^{0'} = [0.875000, 1.000000]$	$\pi_{v_2}^{0'} = 1.0$
$\pi_{v_1}^1 = [0.875000, 1.000000]$	$\pi_{v_2}^1 = 0.0$
$\pi_{v_1}^{1'} = 1.0$	$\pi_{v_2}^{1'} = 1.0$
$\pi_{v_1}^2 = 0.0$	$\pi_{v_2}^2 = 1.0$
$\pi_{v_2}^{2'} = 0.0$	$\pi_{v_1}^{2'} = [0.500000, 1.000000]$
$\pi_{v_1}^3 = [0.500000, 1.000000]$	$\pi_{v_2}^3 = 1.0$
$\pi_{v_1}^{3'} = 0.0$	$\pi_{v_2}^{3'} = 0.0$
$\pi_{v_1}^4 = 1.0$	$\pi_{v_2}^4 = 0.0$

TABLE 9.1 – Valeurs possibles des parties fractionnaires pour le CSP associé à la trace biologique de la boucle négative en condition cyclique. La précision donné à AbSolute pour l’identification de ces valeurs est de 10^{-3} .

9.2 Etude préliminaire de l’influence du nombre d’entité sur le temps de résolution.

Nous avons aussi calculé le temps de résolution du solveur pour des boucles simples de rétrocontrôle négatives de différentes longueurs. Les graphes d’influence sont donc des simples circuits contenant de 2 à 6 entités. Nous avons utilisé le solveur AbSolute pour résoudre les contraintes en faisant varier à la fois le nombre de solutions souhaitées (10, 100, 1000) et la précision (10^{-3} , 10^{-6} et 10^{-9}). Les résultats sont les suivants :

Précision = 10^{-3}				Précision = 10^{-6}				Précision = 10^{-9}			
	10	100	1000		10	100	1000		10	100	1000
2	8	24	208	2	12	28	200	2	12	28	244
3	28	76	556	3	36	84	568	3	44	92	588
4	80	176	1216	4	100	192	1240	4	124	228	1250
5	176	350	2340	5	230	400	2360	5	280	440	2460
6	376	670	3620	6	500	780	3760	6	610	900	3950

TABLE 9.2 – Temps de résolution du solveur (en millisecondes) selon la taille du graphe d’influence représentant une simple boucle de rétrocontrôle négative (de tailles variables), la précision et le nombre de solutions attendues.

9.3 Cycle circadien simplifié en fonction du cycle jour-nuit

Nous avons représenté un cycle circadien simplifié comprenant uniquement les gènes *per* et *cry* ainsi que leurs protéines respectives, voir figure 9.1. Cette boucle négative est influencée par le cycle jour/nuit. La trace biologique utilisée provient des connaissances actuelles sur la synthèse des protéines PER et CRY :

$$\left\{ \begin{array}{l} D_8 \\ H_8 \end{array} \right\} \left(\begin{array}{c} 0.9 \\ \top \\ pc- \end{array} \right); \left(\begin{array}{c} 4.5 \\ \top \\ g+ \end{array} \right); \left(\begin{array}{c} 0.6 \\ \top \\ X+ \end{array} \right); \left(\begin{array}{c} 5.53 \\ \text{slide}^+(g) \\ pc+ \end{array} \right); \left(\begin{array}{c} 0.47 \\ \top \\ L- \end{array} \right); \left(\begin{array}{c} 5.4 \\ \text{slide}^+(pc) \\ g- \end{array} \right); \left(\begin{array}{c} 0.6 \\ \text{slide}^-(L) \\ X- \end{array} \right); \left(\begin{array}{c} 6 \\ \top \\ L+ \end{array} \right) \left\{ \begin{array}{l} D_0 \\ H_0 \end{array} \right\}$$

où $\begin{cases} D_0 \equiv (\eta_g = 0) \wedge (\eta_{pc} = 1) \wedge (\eta_L = 1) \wedge (\eta_X = 0) \\ H_0 \equiv (\pi_g = 0.12) \wedge (\pi_{pc} = 0.12) \wedge (\pi_L = 0) \wedge (\pi_X = 0) \end{cases}$

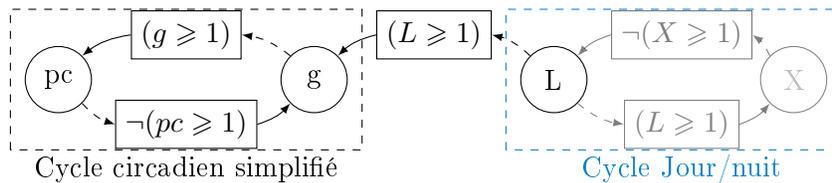


FIGURE 9.1 – Graphe d’influence du cycle circadien simplifié en fonction du cycle jour-nuit. Les gènes *per* et *cry* sont représentés par l’entité *g*, le complexe protéique PER/CRY est représenté par *pc* et la lumière est représentée par *L*. L’entité *X* a été rajoutée pour permettre à la lumière d’osciller sur une période de 24 heures (artefact de simulation).

Nous avons utilisé Holmes BioNet sur ce graphe d’influence et la dynamique de ce graphe est représentée dans la figure 9.2. Le résultat obtenu montre que les gènes *per* et *cry* sont saturés en fin de journée, et le complexe protéique est saturé durant la nuit, similairement aux données biologiques actuelles.

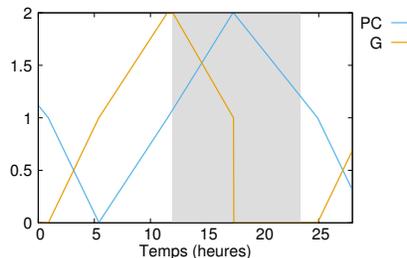


FIGURE 9.2 – Simulation du cycle circadien au cours d’une journée et présentant uniquement le complexe PER/CRY. Le fond blanc représente le jour tandis que le fond grisé représente la nuit.

9.4 Simulation du cycle circadien à 3 entités

Le tableau 9.3 montre l'évolution des entités P , BC , et R du graphe d'influence du cycle circadien. Les résultats obtenus sont en accord avec la trace biologique présentée en sous-section 6.5.1 page 123.

Time	P	BC	R
0.	2.	8.88502383851e-13	1.
2.37192348492e-14	2.	0.	1.
0.0398697068404	1.99348534202	0.	0.
6.12000000002	1.	0.	0.
6.12000000002	1.	0.	0.
6.14666666668	0.	0.00775193798451	0.
9.56000000001	0.	1.	0.
9.56000000001	0.	1.	0.
9.59967479676	0.	2.	0.0162601626016
12.	0.	2.	1.
12.0398697069	0.00651465798044	1.99998531989	2.
18.12	1.	1.99774660313	2.
18.12	1.	1.99774660313	2.
18.9733333334	2.	1.7502435698	2.
21.56	2.	1.	2.
21.56	2.	1.	2.
21.5996747968	2.	0.	1.9837398374
24.	2.	0.	1.
24.	2.	0.	1.
24.0398697069	1.99348534202	0.	0.
30.12	1.	0.	0.
30.12	1.	0.	0.
30.1466666667	0.	0.00775193798451	0.
33.56	0.	1.	0.
33.56	0.	1.	0.
33.5996747968	0.	2.	0.0162601626016
36.	0.	2.	1.
36.0398697069	0.00651465798044	1.99998531989	2.
42.1200000001	1.	1.99774660313	2.
42.1200000001	1.	1.99774660313	2.
42.9733333334	2.	1.7502435698	2.
45.5600000001	2.	1.	2.
45.5600000001	2.	1.	2.
45.5996747968	2.	0.	1.9837398374
48.0000000001	2.	0.	1.

TABLE 9.3 – Evolution des niveaux d'expressions des entités du graphe d'influence du cycle circadien.

9.5 Evolution du graphe d'influence à 5 entités du cycle cellulaire

La figure 9.3 présente l'évolution des entités du graphe cellulaire sur la période de 19 à 23 heures. Les observations de la phase mitotique dans cet intervalle de temps sont en accord avec la trace biologique présentée en section 6.5.2 page 126.

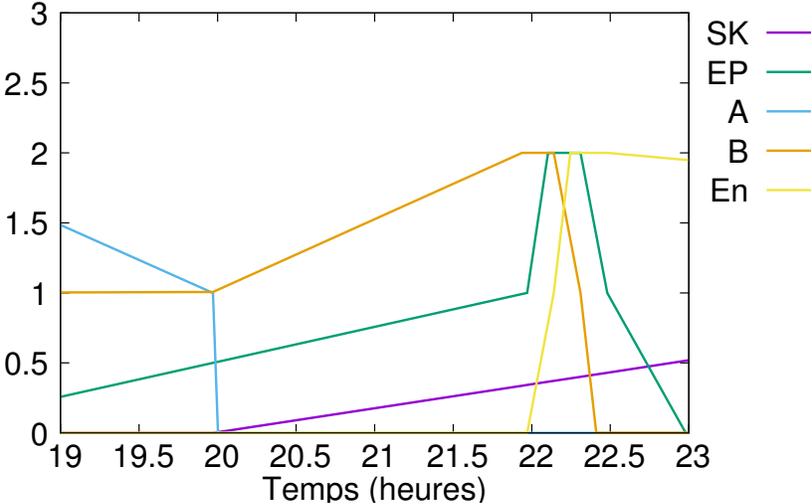


FIGURE 9.3 - Détail de la figure 6.5 sur la période 19h - 23h.

Bibliographie

- [1] F. Agostinelli, N. Ceglia, B. Shahbaba, P. Sassone-Corsi, and P. Baldi. What time is it ? Deep learning approaches for circadian rhythms. *Bioinformatics*, 32(12) :8–17, 2016.
- [2] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell*. Garland Science, 4 edition, 2002.
- [3] H. Alla and R. David. Continuous and hybrid Petri nets. *Journal of Circuits, Systems, and Computers*, 8(01) :159–188, 1998.
- [4] R. Alur, C. Courcoubetis, T. Henzinger, and P. Ho. Hybrid automata : An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid systems*, pages 209–229. Springer, 1993.
- [5] R. Alur and D. Dill. A theory of timed automata. *Theoretical computer science*, 126(2) :183–235, 1994.
- [6] R. Anafi, Y. Lee, T. Sato, A. Venkataraman, C. Ramanathan, I. Kavakli, M. Hughes, J. Baggs, J. Grove, and A. Liu. Machine learning helps identify chrono as a circadian clock component. *PLoS biology*, 12(4) :1–18, 2014.
- [7] E. Asarin, O. Maler, and A. Pnueli. Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoretical computer science*, 138(1) :35–65, 1995.
- [8] E. Asarin and G. Schneider. Widening the boundary between decidable and undecidable hybrid systems. In *International Conference on Concurrency Theory*, pages 193–208. Springer, 2002.
- [9] S. Banin, L. Moyal, S.-Y. Shieh, Y. Taya, C. Anderson, L. Chessa, N. Smorodinsky, C. Prives, Y. Reiss, Y. Shiloh, et al. Enhanced phosphorylation of p53 by atm in response to dna damage. *Science*, 281(5383) :1674–1677, 1998.
- [10] D. Barik, D. Ball, J. Peccoud, and J. Tyson. A stochastic model of the yeast cell cycle reveals roles for feedback regulation in limiting cellular variability. *PLoS computational biology*, 12(12), 2016.

BIBLIOGRAPHIE

- [11] G. Batt, H. De Jong, J. Geiselman, and M. Page. Analysis of genetic regulatory networks : A model-checking approach. In *Proc. 17th Int. Work. Qualitative Reasoning (QR-03)*, pages 31–38, 2003.
- [12] G. Batt, S. Soliman, C. Chaouiya, F. Fages, E. Remy, L. Calzone, and Pommereau. F. Modelling molecular networks : relationships between different formalisms and levels of details. 2010.
- [13] J. Behaegel. A l’heure des super médicaments. https://youtu.be/y3sFxj0_P845, 2017.
- [14] J. Behaegel, J.-P. Comet, and M. Folschette. Constraint identification using modified Hoare logic on hybrid models of gene networks. In *LIPICs-Leibniz International Proceedings in Informatics*, volume 90. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.
- [15] J. Bengtsson, K. Larsen, F. Larsson, P. Pettersson, and W. Yi. Uppaal—a tool suite for automatic verification of real-time systems. In *International Hybrid Systems Workshop*, pages 232–243. Springer, 1995.
- [16] F. Benhamou, F. Goualard, L. Granvilliers, and J.-F. Puget. Revising hull and box consistency. In *Logic Programming : The 1999 International Conference, Las Cruces, New Mexico, USA, November 29 - December 4, 1999*, pages 230–244, 1999.
- [17] G. Bernot, J.-P. Comet, Z. Khalis, A. Richard, and O. F. Roux. A genetically modified Hoare logic. *Theoretical Computer Science*, 2018. <https://doi.org/10.1016/j.tcs.2018.02.003>.
- [18] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. Application of formal methods to biological regulatory networks : extending Thomas’ asynchronous logical approach with temporal logic. *Journal of theoretical biology*, 229(3) :339–347, 2004.
- [19] B. Berthomieu and M. Diaz. Modeling and verification of time dependent systems using time Petri nets. *IEEE transactions on software engineering*, 17(3) :259–273, 1991.
- [20] C. Bessiere. Constraint propagation. In *Handbook of Constraint Programming*, pages 29–83. 2006.
- [21] T. Bhola, O. Kapuy, and P. Vinod. Computational modelling of meiotic entry and commitment. *Scientific reports*, 8(1) :180–190, 2018.
- [22] J. Bieler, R. Cannavo, K. Gustafson, C. Gobet, D. Gatfield, and F. Naef. Robust synchronization of coupled circadian and cell cycle oscillators in single mammalian cells. *Molecular systems biology*, 10(7), 2014.

-
- [23] J. Bischoff, P. Friedman, D. Marshak, C. Prives, and D. Beach. Human p53 is phosphorylated by p60-cdc2 and cyclin B-cdc2. *Proceedings of the National Academy of Sciences*, 87(12) :4766–4770, 1990.
- [24] G. Bjarnason, R. Jordan, and R. Sothorn. Circadian variation in the expression of cell-cycle proteins in human oral epithelium. *The American journal of pathology*, 154(2) :613–622, 1999.
- [25] I. Blomberg and I. Hoffmann. Ectopic expression of Cdc25a accelerates the G1/S transition and leads to premature activation of cyclin E-and cyclin A-dependent kinases. *Molecular and cellular biology*, 19(9) :6183–6194, 1999.
- [26] J. Boonstra and J.-A. Post. Molecular events associated with reactive oxygen species and cell cycle progression in mammalian cells. *Gene*, 337 :1–13, 2004.
- [27] A. Bugge, D. Feng, L. Everett, E. Briggs, S. Mullican, F. Wang, J. Jager, and M. Lazar. Rev-erba and Rev-erb β coordinately protect the circadian clock and normal metabolic function. *Genes & development*, 26(7) :657–667, 2012.
- [28] C. Canman, D.-S. Lim, K. Cimprich, Y. Taya, K. Tamai, K. Sakaguchi, E. Appella, M. Kastan, and J. Siliciano. Activation of the ATM kinase by ionizing radiation and phosphorylation of p53. *Science*, 281(5383) :1677–1679, 1998.
- [29] A. Cashmore, J. Jarillo, Y.-J. Wu, and D. Liu. Cryptochromes : blue light receptors for plants and animals. *Science*, 284(5415) :760–765, 1999.
- [30] A. Chaix, A. Zarrinpar, and S. Panda. The circadian coordination of cell biology. *J Cell Biol*, 215(1) :15–25, 2016.
- [31] C. Chaouiya. Petri net modelling of biological networks. *Briefings in bioinformatics*, 8(4) :210–219, 2007.
- [32] C. Chaouiya, A. Naldi, E. Remy, and D. Thieffry. Petri net representation of multi-valued logical regulatory graphs. *Natural Computing*, 10(2) :727–750, 2011.
- [33] J. Clairambault. Physiologically based modelling of circadian control on cell proliferation. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pages 173–176. IEEE, 2006.
- [34] A. Colmerauer and P. Roussel. The birth of prolog. In *History of programming languages—II*, pages 331–367. ACM, 1996.
- [35] J.-P. Comet, G. Bernot, A. Das, F. Diener, C. Massot, and A. Cessieux. Simplified models for the mammalian circadian clock. *Procedia Computer Science*, 11 :127–138, 2012.
- [36] F. Corblin, E. Fanchon, and L. Trilling. Applications of a formal approach to decipher discrete genetic networks. *BMC bioinformatics*, 11(1) :385–405, 2010.

BIBLIOGRAPHIE

- [37] E. Cornillon. *Modèles qualitatifs de réseaux génétiques : réduction de modèles et introduction d'un temps continu*. PhD thesis, Université Côte d'Azur, 2017.
- [38] P. Cousot and R. Cousot. Abstract interpretation : a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 238–252. ACM, 1977.
- [39] H. De Jong. Modeling and simulation of genetic regulatory systems : a literature review. *Journal of computational biology*, 9(1) :67–103, 2002.
- [40] E. Dijkstra. Guarded commands, nondeterminacy and formal derivation of programs. *Commun. ACM*, 18(8) :453–457, 1975.
- [41] R. El Cheikh, S. Bernard, and N. El Khatib. A multiscale modelling approach for the regulation of the cell cycle by the circadian clock. *Journal of theoretical biology*, 426 :117–125, 2017.
- [42] P. Emery, V. So, M. Kaneko, J. Hall, and M. Rosbash. Cry, a drosophila clock and light-regulated cryptochrome, is a major contributor to circadian rhythm resetting and photosensitivity. *Cell*, 95(5) :669–679, 1998.
- [43] Commission Européenne. Communication de la commission au parlement européen et au conseil concernant l'interdiction de l'expérimentation animale et l'interdiction de mise sur le marché dans le secteur des cosmétiques et faisant le point sur les méthodes de substitution à l'expérimentation animale. <https://eur-lex.europa.eu/legal-content/FR/TXT/?uri=celex:52013DC0135>, 2013.
- [44] T. Evans, E. Rosenthal, J. Youngblom, D. Distel, and T. Hunt. Cyclin : a protein specified by maternal mRNA in sea urchin eggs that is destroyed at each cleavage division. *Cell*, 33(2) :389–396, 1983.
- [45] F. Fages, S. Soliman, and N. Chabrier-Rivier. Modelling and querying interaction networks in the biochemical abstract machine biocham. *Journal of Biological Physics and Chemistry*, 4 :64–73, 2004.
- [46] A. Fauré, A. Naldi, C. Chaouiya, and D. Thieffry. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22(14) :124–131, 2006.
- [47] C. Feillet, P. Krusche, F. Tamanini, R. Janssens, M. Downey, P. Martin, M. Teboul, S. Saito, F. Lévi, T. Bretschneider, et al. Phase locking and multiple oscillating attractors for the coupled mammalian clock and cell cycle. *Proceedings of the National Academy of Sciences*, 111(27) :9828–9833, 2014.

-
- [48] J. Ferrell, T. Tsai, and Q. Yang. Modeling the cell cycle : why do certain circuits oscillate ? *Cell*, 144(6) :874–885, 2011.
- [49] D. Fisher, L. Krasinska, D. Coudreuse, and B. Novák. Phosphorylation network dynamics in the control of cell cycle transitions. *Journal of cell science*, 125(20) :4703–4711, 2012.
- [50] W. Flemming. *Zellsubstanz, kern und zelltheilung*. Vogel, 1882.
- [51] G. Gardey, D. Lime, M. Magnin, et al. Romeo : A tool for analyzing time Petri nets. In *International Conference on Computer Aided Verification*, pages 418–423. Springer, 2005.
- [52] L. Geitler. Die entstehung der polyploiden somakerne der heteropteren durch chromosomenteilung ohne kernteilung. *Zeitschrift für Zellforschung und Mikroskopische Anatomie Abt. B Chromosoma*, 1(1) :1–22, 1939.
- [53] L. Geitler. *Endomitose und endomitotische Polyploidisierung*, volume 6. Springer-Verlag, 2013.
- [54] C. Gérard and A. Goldbeter. Temporal self-organization of the Cyclin/Cdk network driving the mammalian cell cycle. *Proceedings of the National Academy of Sciences*, 106(51) :21643–21648, 2009.
- [55] C. Gérard and A. Goldbeter. Entrainment of the mammalian cell cycle by the circadian clock : modeling two coupled cellular rhythms. *PLoS computational biology*, 8(5) :1–21, 2012.
- [56] C. Gérard, J. Tyson, D. Coudreuse, B. Novák, and J. Saucerman. Cell cycle control by a minimal Cdk network. *PLoS Comput Biol*, 11(2) :1–27, 2015.
- [57] L. Glass and S. Kauffman. Co-operative components, spatial localization and oscillatory cellular dynamics. *Journal of theoretical biology*, 34(2) :219–237, 1972.
- [58] T. Gotoh, J. Kim, J. Liu, M. Vila-Caballer, P. Stauffer, J. Tyson, and C. Finkielstein. Model-driven experimental approach reveals the complex regulatory distribution of p53 by the circadian factor period 2. *Proceedings of the National Academy of Sciences*, 113(47) :13516–13521, 2016.
- [59] A. Gréchez-Cassiau, B. Rayet, F. Guillaumond, M. Teboul, and F. Delaunay. The circadian clock component BMAL1 is a critical regulator of p21WAF1/CIP1 expression and hepatocyte proliferation. *Journal of Biological Chemistry*, 283(8) :4535–4542, 2008.
- [60] M. L. Guerriero, A. Pokhilko, A. P. Fernández, K. Halliday, A. Millar, and J. Hillston. Stochastic properties of the plant circadian clock. *Journal of The Royal Society Interface*, pages 1–13, 2011.
- [61] L. Hartwell and M. Kastan. Cell cycle control and cancer. *Science*, 266(5192) :1821–1828, 1994.

BIBLIOGRAPHIE

- [62] M. Heiner, D. Gilbert, and R. Donaldson. Petri nets for systems and synthetic biology. In *International school on formal methods for the design of computer, communication and software systems*, pages 215–264. Springer, 2008.
- [63] T. Henzinger. The theory of hybrid automata. In *Verification of Digital and Hybrid Systems*, pages 265–292. Springer, 2000.
- [64] T. Henzinger, P.-H. Ho, and H. Wong-Toi. Hytech : A model checker for hybrid systems. *International Journal on Software Tools for Technology Transfer*, 1(1-2) :110–122, 1997.
- [65] T. Henzinger, P. Kopke, A. Puri, and P. Varaiya. What’s decidable about hybrid automata ? *Journal of Computer and System Sciences*, 57(1) :94–124, 1998.
- [66] A. Hinton, M. Kwiatkowska, Gethin D. Kwiatkowska, N., and D. Parker. Prism : A tool for automatic verification of probabilistic systems. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 441–444. Springer, 2006.
- [67] T. Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10) :576–580, 1969.
- [68] J. Hogenesch, W. Chan, V. Jackiw, C. Brown, Y.-Z. Gu, M. Pray-Grant, G. Perdew, and C. Bradfield. Characterization of a subset of the basic-helix-loop-helix-pas superfamily that interacts with components of the dioxin signaling pathway. *Journal of Biological Chemistry*, 272(13) :8581–8593, 1997.
- [69] J. Hooman. Extending Hoare logic to real-time. *Formal Aspects of Computing*, 6(1) :801–825, 1994.
- [70] W. Huang, K. M. Ramsey, B. Marcheva, and J. Bass. Circadian rhythms, sleep, and metabolism. *The Journal of clinical investigation*, 121(6) :2133–2141, 2011.
- [71] T. Hunt and P. Sassone-Corsi. Riding tandem : circadian clocks and the cell cycle. *Cell*, 129(3) :461–464, 2007.
- [72] M. Ikeda and M. Nomura. cDNA cloning and tissue-specific expression of a novel basic helix–loop–helix/PAS protein (BMAL1) and identification of alternatively spliced variants with alternative translation initiation site usage. *Biochemical and biophysical research communications*, 233(1) :258–264, 1997.
- [73] M. Ishiura, S. Kutsuna, S. Aoki, H. Iwasaki, C. Andersson, A. Tanabe, S. Golden, C. Johnson, and T. Kondo. Expression of a gene cluster kaiABC as a circadian feedback process in cyanobacteria. *Science*, 281(5382) :1519–1523, 1998.
- [74] S. Jamshidi. *Comparing discrete, continuous and hybrid modelling approaches of gene regulatory networks*. PhD thesis, Freie Universität Berlin, 2013.

- [75] O. Kapuy, P. Vinod, G. Bánhegyi, and B. Novák. Systems-level feedback regulation of cell cycle transitions in *ostreococcus tauri*. *Plant Physiology and Biochemistry*, 126 :39–46, 2018.
- [76] S. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3) :437–467, 1969.
- [77] Z. Khalis, J.-P. Comet, A. Richard, and G. Bernot. The SMBioNet method for discovering models of gene regulatory networks. *Genes, genomes and genomics*, 3(1) :15–22, 2009.
- [78] K. K. Khanna, K. Keating, S. Kozlov, S. Scott, M. Gatei, K. Hobson, Y. Taya, B. Gabrielli, D. Chan, S. Lees-Miller, et al. Atm associates with and phosphorylates p53 : mapping the region of interaction. *Nature genetics*, 20(4) :398–401, 1998.
- [79] K. Kohn. Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Molecular biology of the cell*, 10(8) :2703–2734, 1999.
- [80] N. Koike, S.-H. Yoo, H.-C. Huang, V. Kumar, C. Lee, T.-K. Kim, and J. Takahashi. Transcriptional architecture and chromatin landscape of the core circadian clock in mammals. *Science*, 338(6105) :349–354, 2012.
- [81] R. Konopka and S. Benzer. Clock mutants of *drosophila melanogaster*. *Proceedings of the National Academy of Sciences*, 68(9) :2112–2116, 1971.
- [82] K. Lamia, U. Sachdeva, L. DiTacchio, E. Williams, J. Alvarez, D. Egan, D. Vasquez, H. Juguilon, S. Panda, R. Shaw, et al. Ampk regulates the circadian clock by cryptochrome phosphorylation and degradation. *Science*, 326(5951) :437–440, 2009.
- [83] S. Larochelle, K. Merrick, M.-E. Terret, L. Wohlbold, N. Barboza, C. Zhang, K. Shokat, P. Jallepalli, and R. Fisher. Requirements for Cdk7 in the assembly of Cdk1/cyclin B and activation of Cdk2 revealed by chemical genetics in human cells. *Molecular cell*, 25(6) :839–850, 2007.
- [84] K. Larsen, P. Pettersson, and W. Yi. Uppaal in a nutshell. *International journal on software tools for technology transfer*, 1(1-2) :134–152, 1997.
- [85] M.-C. Lebart and J. Mariani. La régulation du cycle cellulaire. <http://www.snv.jussieu.fr/bmedia/cyclecellBM/>, 2004. [Mis en ligne et modification - Juin 2004].
- [86] T. Lehouillier, F. Soumis, J. Omer, and C. Allignol. Measuring the interactions between air traffic control and flow management using a simulation-based framework. *Computers & Industrial Engineering*, 99 :269–279, 2016.
- [87] F. Levi, R. Zidani, and J.-L. Misset. Randomised multicentre trial of chronotherapy with oxaliplatin, fluorouracil, and folinic acid in metastatic colorectal cancer. *The Lancet*, 350(9079) :681–686, 1997.

BIBLIOGRAPHIE

- [88] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, 101(14) :4781–4786, 2004.
- [89] L. Ljung. System identification. In *Signal analysis and prediction*, pages 163–173. Springer, 1998.
- [90] V. Marek and M. Truszczyński. Stable models and an alternative logic programming paradigm. In *The Logic Programming Paradigm*, pages 375–398. Springer, 1999.
- [91] H. Matsuno, Y. Tanaka, H. Aoshima, M. Matsui, S. Miyano, et al. Biopathways representation and simulation on hybrid functional Petri net. *In silico biology*, 3(3) :389–404, 2003.
- [92] T. Matsuo, S. Yamaguchi, S. Mitsui, A. Emi, F. Shimoda, and H. Okamura. Control mechanism of the circadian clock for timing of cell division in vivo. *Science*, 302(5643) :255–259, 2003.
- [93] T. Miki, T. Matsumoto, Z. Zhao, and C. C. Lee. p53 regulates Period2 expression and the circadian clock. *Nature communications*, 4, 2013.
- [94] R. Moore. Interval arithmetic, 1966.
- [95] R. Moore and V. Eichler. Loss of a circadian adrenal corticosterone rhythm following suprachiasmatic lesions in the rat. *Brain research*, 1972.
- [96] M. Myers, K. Wager-Smith, A. Rothenfluh-Hilfiker, and M. Young. Light-induced degradation of timeless and entrainment of the drosophila circadian clock. *Science*, 271(5256) :1736–1740, 1996.
- [97] B. Neveu and G. Trombettoni. Acid : Disjonction constructive adaptative sur intervalles. In *Dixièmes Journées Francophones de Programmation par Contraintes (JFPC 2014)*, 2014.
- [98] J. Ninin. Global optimization based on contractor programming : An overview of the IBEX library. In *Mathematical Aspects of Computer and Information Sciences - 6th International Conference, MACIS 2015, Berlin, Germany, November 11-13, 2015, Revised Selected Papers*, pages 555–559, 2015.
- [99] B. Novak and J. Tyson. A model for restriction point control of the mammalian cell cycle. *Journal of theoretical biology*, 230(4) :563–579, 2004.
- [100] B. Novak, J. Tyson, B. Gyorffy, and A. Csikasz-Nagy. Irreversible cell-cycle transitions are due to systems-level feedback. *Nature cell biology*, 9(7–11) :724, 2007.
- [101] P. Nurse. Ordering S phase and M phase in the cell cycle. *Cell*, 79(4) :547–550, 1994.

- [102] A. Palinkas and A. Bockmayr. Petri nets for integrated models of metabolic and gene regulatory networks. *Workshop on Constraint Based Methods for Bioinformatics*, pages 45–57, 2011.
- [103] M. Pelleau, A. Miné, C. Truchet, and F. Benhamou. A constraint solver based on abstract domains. In *Verification, Model Checking, and Abstract Interpretation, 14th International Conference, VMCAI 2013, Rome, Italy, January 20-22, 2013. Proceedings*, pages 434–454, 2013.
- [104] O. Perriquet and P. Barahona. Constraint-based strategy for pairwise RNA secondary structure prediction. In *Portuguese Conference on Artificial Intelligence*, pages 86–97. Springer, 2009.
- [105] J.-M. Peters. The anaphase promoting complex/cyclosome : a machine designed to destroy. *Nature reviews Molecular cell biology*, 7(9), 2006.
- [106] M. Pisu, A. Concas, and G. Cao. A novel quantitative model of cell cycle progression based on cyclin-dependent kinases activity and population balances. *Computational biology and chemistry*, 55 :1–13, 2015.
- [107] A. Platzer. Differential dynamic logic for hybrid systems. *Journal of Automated Reasoning*, 41(2) :143–189, 2008.
- [108] Z. Qu, W. R. MacLellan, and J. Weiss. Dynamics of the cell cycle : checkpoints, sizers, and timers. *Biophysical journal*, 85(6) :3600–3611, 2003.
- [109] G. Rey, F. Cesbron, J. Rougemont, H. Reinke, M. Brunner, and F. Naef. Genome-wide and phase-specific DNA-binding rhythms of BMAL1 control circadian output functions in mouse liver. *PLoS biology*, 9(2) :1–18, 2011.
- [110] S. Risau-Gusman and P. Gleiser. Modelling the effect of phosphorylation on the circadian clock of drosophila. *Journal of theoretical biology*, 307 :53–61, 2012.
- [111] D. Santamaría, C. Barrière, A. Cerqueira, S. Hunt, C. Tardy, K. Newton, J. Cáceres, P. Dubus, M. Malumbres, and M. Barbacid. Cdk1 is sufficient to drive the mammalian cell cycle. *Nature*, 448(7155), 2007.
- [112] T. Sato, S. Panda, L. Miraglia, T. Reyes, R. Rudic, P. McNamara, K. Naik, G. FitzGerald, S. Kay, and J. Hogenesch. A functional genomics strategy reveals Rora as a component of the mammalian circadian clock. *Neuron*, 43(4) :527–537, 2004.
- [113] A. Satyanarayana and P. Kaldis. Mammalian cell-cycle regulation : several Cdks, numerous cyclins and diverse compensatory mechanisms. *Oncogene*, 28(33) :2925–2939, 2009.
- [114] W. Sedgwick, T. Thompson, and W. Harry. *A short history of science*. Macmillan, 1917.

BIBLIOGRAPHIE

- [115] J. Selfridge, T. Gotoh, S. Schiffhauer, J. Liu, P. Stauffer, A. Li, D. Capelluto, and C. Finkielstein. Chronotherapy : intuitive, sound, founded... but not broadly applied. *Drugs*, 76(16) :1507–1521, 2016.
- [116] R. Sheaff, M. Groudine, M. Gordon, J. Roberts, and B. Clurman. Cyclin E-CDK2 is a regulator of p27Kip1. *Genes & development*, 11(11) :1464–1478, 1997.
- [117] C. Sherr and J. Roberts. Cdk inhibitors : positive and negative regulators of G1-phase progression. *Genes & development*, 13(12) :1501–1512, 1999.
- [118] H. Siebert and A. Bockmayr. Temporal constraints in the logical analysis of regulatory networks. *Theoretical Computer Science*, 391(3) :258–275, 2008.
- [119] R. Singhanian, R. M. Sramkoski, J. Jacobberger, J. Tyson, and D. Beard. A hybrid model of mammalian cell cycle regulation. *PLoS Comput Biol*, 7(2) :1–10, 2011.
- [120] E. H. Snoussi. Qualitative dynamics of piecewise-linear differential equations : a discrete mapping approach. *Dynamics and stability of Systems*, 4(3-4) :565–583, 1989.
- [121] S. Spencer, S. Cappell, F.-C. Tsai, K. W. Overton, C. Wang, and T. Meyer. The proliferation-quiescence decision is controlled by a bifurcation in CDK2 activity at mitotic exit. *Cell*, 155(2) :369–383, 2013.
- [122] T. Stauner, O. Müller, and M. Fuchs. Using hytech to verify an automotive control system. In *International Workshop on Hybrid and Real-Time Systems*, pages 139–153. Springer, 1997.
- [123] J. Takahashi. Transcriptional architecture of the mammalian circadian clock. *Nature Reviews Genetics*, 18(3), 2017.
- [124] R. Thomas. Boolean formalization of genetic control circuits. *Journal of theoretical biology*, 42(3) :563–585, 1973.
- [125] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. ii. logical analysis of regulatory networks in terms of feedback circuits. *Chaos : An Interdisciplinary Journal of Nonlinear Science*, 11(1) :180–195, 2001.
- [126] F. Toyoshima-Morimoto, E. Taniguchi, and E. Nishida. Plk1 promotes nuclear translocation of human Cdc25c during prophase. *EMBO reports*, 3(4) :341–348, 2002.
- [127] P. Traynard, C. Feillet, S. Soliman, F. Delaunay, and F. Fages. Model-based investigation of the circadian clock and cell cycle coupling in mouse embryonic fibroblasts : Prediction of RevErb- α up-regulation during mitosis. *Biosystems*, 2016.
- [128] G. Trombettoni, I. Araya, B. Neveu, and G. Chabert. Inner regions and interval linearizations for global optimization. In *Proceedings of the Twenty-Fifth AAAI Conference on*

-
- Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011, 2011.*
- [129] G. Trombettoni and G. Chabert. Constructive interval disjunction. In *Principles and Practice of Constraint Programming - CP 2007, 13th International Conference, CP 2007, Providence, RI, USA, September 23-27, 2007, Proceedings*, pages 635–650, 2007.
- [130] K. Tsumoto, G. Kurosawa, T. Yoshinaga, and K. Aihara. Modeling light adaptation in circadian clock : Prediction of the response that stabilizes entrainment. *PLoS One*, 6(6) :1–12, 2011.
- [131] K. Tsumoto, T. Yoshinaga, H. Iida, H. Kawakami, and K. Aihara. Bifurcations in a mathematical model for circadian oscillations of clock genes. *Journal of Theoretical Biology*, 239(1) :101–122, 2006.
- [132] A. Tyner and A. Gartel. Roles of cyclin kinase inhibitors in G1 phase progression. *G1 phase progression. Kluwer, New York*, pages 58–76, 2003.
- [133] J. Tyson and B. Novak. Temporal organization of the cell cycle. *Current Biology*, 18(17) :R759–R768, 2008.
- [134] C. Valouxis and E. Housos. Constraint programming approach for school timetabling. *Computers & Operations Research*, 30(10) :1555–1572, 2003.
- [135] A. Vidal and A. Koff. Cell-cycle inhibitors : three families united by a common cause. *Gene*, 247(1) :1–15, 2000.
- [136] S. Videla, C. Guziolowski, F. Eduati, S. Thiele, M. Gebser, J. Nicolas, J. Saez-Rodriguez, T. Schaub, and A. Siegel. Learning Boolean logic models of signaling networks with ASP. *Theoretical Computer Science*, 599 :79–101, 2015.
- [137] E. Vigo, H. Müller, E. Prosperini, G. Hateboer, P. Cartwright, M. C. Moroni, and K. Helin. CDC25A phosphatase is a target of E2F and is required for efficient E2F-induced S phase. *Molecular and cellular biology*, 19(9) :6379–6395, 1999.
- [138] M. Vitaterna, D. King, A.-M. Chang, J. Kornhauser, P. Lowrey, J. McDonald, W. Dove, L. Pinto, F. Turek, and J. Takahashi. Mutagenesis and mapping of a mouse gene, clock, essential for circadian behavior. *Science*, 264(5159) :719–725, 1994.
- [139] A. Winfree. Biological rhythms and the behavior of populations of coupled oscillators. *Journal of theoretical biology*, 16(1) :15–42, 1967.
- [140] A. Woller, H. Duez, B. Staels, and M. Lefranc. A mathematical model of the liver circadian clock linking feeding and fasting cycles to clock function. *Cell Reports*, 17(4) :1087–1097, 2016.

- [141] Z. Xie and D. Kulasiri. Modelling of circadian rhythms in drosophila incorporating the interlocked PER/TIM and VRI/PDP1 feedback loops. *Journal of theoretical biology*, 245(2) :290–304, 2007.
- [142] Q. Yang and J. Ferrell. The Cdk1–APC/C cell cycle oscillator circuit functions as a time-delayed, ultrasensitive switch. *Nature cell biology*, 15(5) :1–13, 2013.
- [143] Q. Yang, B. Pando, G. Dong, S. Golden, and A. Van Oudenaarden. Circadian gating of the cell cycle revealed in single cyanobacterial cells. *Science*, 327(5972) :1522–1526, 2010.
- [144] S.-H. Yoo, S. Yamazaki, P. Lowrey, K. Shimomura, C. Ko, E. Buhr, S. Siepk, H.-K. Hong, W. Oh, and O. Yoo. PERIOD2 :LUCIFERASE real-time reporting of circadian dynamics reveals persistent circadian oscillations in mouse peripheral tissues. *Proceedings of the National Academy of Sciences of the United States of America*, 101(15) :5339–5346, 2004.
- [145] R. Zhang, N. Lahens, H. Ballance, M. Hughes, and J. Hogenesch. A circadian gene expression atlas in mammals : implications for biology and medicine. *Proceedings of the National Academy of Sciences*, 111(45) :16219–16224, 2014.
- [146] X. Zhao, T. Hirota, X. Han, H. Cho, L.-W. Chong, K. Lamia, S. Liu, A. Atkins, E. Bannayo, C. Liddle, et al. Circadian amplitude regulation via FBXW7-targeted REV-ERB α degradation. *Cell*, 165(7) :1644–1657, 2016.

Index

A	
AbSolute	103
Anaphase	23
APC	29
Atome de chemin discret	72
Atteignabilité	62
Automate hybride	14, 59
Automate rectangulaire	62
B	
BMAL1	31
C	
Cadre de modélisation	13, 33
Cadre différentiel	13, 34
Cadre hybride	13, 34
Cadre qualitatif	13, 34
Cadre stochastique	13, 34
Cadres de modélisation discrets	43
Cadres de modélisation hybrides	44
Cdk	24
Célérités	16, 49
Centromère	21
Centrosome	21
Checkpoints	24
Chemins élémentaires	71
Chromatide	21
CKI	25
CLOCK	31
Complétude	68
Condition discrète	69
Condition hybride	69
Correction	68
Couple d’assertion	72, 73
CRY	31
CSP	97
Cycle cellulaire	15, 20
Cycle circadien	16, 30
Cycle limite	57
Cycle nyctéméral	16
Cycline	25
Cycline A	27
Cycline B	28
Cycline D	27
Cycline E	27
Cytodiérèse	23
D	
Délai de contact	50
DNA Damage Checkpoint	24
E	
Endomitose	23
Endoréplication	23
Entité glissante	52
Etat hybride	49

Etat qualitatif	46
Etat voisin	46

F

Facteur de transcription	20
Facteurs de croissance	27
<i>first</i>	53

G

Gène	19
Graphe d'états	46
Graphe d'influence	12, 33

H

Holmes BioNet	113
Horloge centrale	30

I

Ibex	102
Interphase	20, 21

J

Jonctions continues	79
Jonctions discrètes	79

K

Kinétochores	22
--------------------	----

L

Langage d'assertion	71
Langage de chemins	70
Langage de propriétés	69
Logique de Hoare	14, 66
Logique de Hoare hybride	69

M

Métaphase	23
Méthodes formelles	14, 61
Mitose	20

Mitotic Checkpoint	24
--------------------------	----

Modélisation	12
--------------------	----

Multiplexe	46
------------------	----

Mur externe	52
-------------------	----

Mur interne	52
-------------------	----

P

p53	26
-----------	----

Paramètres dynamiques	33
-----------------------------	----

Partie fractionnaire	49
----------------------------	----

PER	31
-----------	----

Phase mitotique	22
-----------------------	----

Plus faible précondition	76
--------------------------------	----

Point fixe	100
------------------	-----

Prédécesseurs	47
---------------------	----

Programmation par contraintes	98
-------------------------------------	----

Prométaphase	22
--------------------	----

Propagateurs	100
--------------------	-----

Prophase	22
----------------	----

Q

Quiescence	20, 22
------------------	--------

R

Réécriture symbolique	104
-----------------------------	-----

Règles d'inférence	82
--------------------------	----

Replication Checkpoint	24
------------------------------	----

Réseau de régulation	12
----------------------------	----

Ressource	47
-----------------	----

Rythme circadien	30
------------------------	----

S

Sémantique de transformation de prédicats 14, 67	
---	--

Sillon de division	23
--------------------------	----

Solveurs de contraintes	15, 98
-------------------------------	--------

Stratégie backward	67
--------------------------	----

Système de coordination circadienne	30
--	----

T

Télophase	23
Termes continus	69
Termes discrets	69
Trace biologique	16, 67

Transition continue	53
---------------------------	----

Transition discrète	53
---------------------------	----

Triplet de Hoare	14, 66
------------------------	--------

V

Vérification formelle	14
-----------------------------	----