

Méthodes de vérification formelle pour la modélisation en biologie: le cas des réseaux de régulation biologique

Gilles BERNOT, Hélène COLLAVIZZA, Jean-Paul COMET



Université Côte d'Azur, laboratoire I3S, UMR CNRS 7271,
CS 40121, 06903 Sophia Antipolis Cedex, France

Résumé. On ne compte plus le nombre de fois où les chercheurs en méthodes formelles pour le génie logiciel ont entendu cette plaisanterie: « Les méthodes formelles ont toujours été l'avenir de l'informatique... et le resteront toujours! ». Dans ce chapitre nous montrons que les méthodes formelles sont le présent de l'analyse des réseaux de régulation en biologie des systèmes.

Plus précisément, nous donnons les définitions symboliques de deux formalismes: celui de la théorie qualitative de modélisation des réseaux de régulation biologique de René Thomas, et celui de la modélisation hybride, avec délais, de ces réseaux. Nous décrivons pour chacun des deux formalismes une méthodologie complète d'aide à la modélisation fondée sur les méthodes de vérification formelle. On se focalise sur une adaptation aux réseaux de régulation des techniques de validation par model-checking et de preuve en logique de Hoare, mais bien d'autres méthodes formelles sont applicables en modélisation pour la biologie. Nous montrons ainsi que les approches logiques classiques du génie logiciel s'adaptent remarquablement efficacement au problème majeur de la modélisation des systèmes complexes: l'identification des paramètres.

1.1. Introduction

La biologie des systèmes a pour but l'étude des processus biologiques dont le comportement n'est pas facilement déductible des interactions que les experts connaissent. C'est en ce sens que ces systèmes sont considérés comme des systèmes complexes: la connaissance des interactions locales ne suffit pas à la compréhension du système dans son ensemble. La biologie des systèmes tente donc d'apporter une vision intégrée du fonctionnement de ces systèmes intriqués.

Les réseaux de régulation biologique (aux différents niveaux d'organisation du vivant) gouvernent pour une large part le comportement des systèmes biologiques et une modélisation mathématique et informatique est indispensable pour comprendre leur fonctionnement, élucider les liens de causalité afin de, par exemple, modifier des fonctions biologiques, concevoir des traitements pharmaceutiques, optimiser la production de certaines molécules à des fins de biologie synthétique, *etc.* Parmi les quelques techniques qui ont démontré leur utilité pour traiter ce type de systèmes complexes, les méthodes formelles, fortement développées en informatique (D'Silva *et al.* (2008)), se sont avérées très efficaces pour accompagner ces questions pluridisciplinaires.

Dans ce chapitre, nous illustrons l'apport majeur des méthodes formelles à la modélisation des systèmes biologiques complexes en choisissant de se focaliser sur les réseaux de régulation. D'un point de vue structurel, le système est composé d'un ensemble d'entités qui sont interdépendantes et on connaît certaines des interactions que l'on peut qualifier d'activation ou d'inhibition. Cependant, la combinatoire des influences croisées n'est quasiment jamais explorée, et c'est cette combinatoire qui participe à la richesse des comportements observés. Les méthodes de vérification formelle peuvent ici être d'une grande aide pour faire émerger les comportements globaux qui sont compatibles avec les connaissances locales et souvent partielles.

Le chapitre s'organise de la manière suivante. Nous présentons en section 1.2 le cadre de modélisation des réseaux de régulation biologique introduit par René Thomas, voir par exemple Thomas (1978), Thomas and d'Ari (1990) ou Thomas (1991*b*), dans lequel il faut bien distinguer le graphe d'influences qui décrit la partie structurelle du système biologique et la dynamique qu'on peut lui associer et qui est aussi représentée par un graphe appelé graphe de transitions. Ce cadre de modélisation fait l'hypothèse que toutes les interactions peuvent être « seuillées », ce qui permet de discrétiser les espaces de concentrations tout en gardant une cohérence de la dynamique, Cf. Snoussi (1989). De plus, ce cadre de modélisation se focalise plus sur les successions des événements plutôt que sur les durées qui séparent les événements. Autrement dit, le temps mesurable par une horloge (*chronométrie*) a été abstrait à la simple succession des événements (*chronologie*). La dynamique d'un tel réseau est contrôlée par un ensemble de paramètres (jouant le rôle des constantes cinétiques) et la grande difficulté de l'activité de modélisation réside dans la

recherche des paramétrages qui mènent à une dynamique cohérente avec les connaissances biologiques et les données expérimentales (Snoussi and Thomas (1993)).

La section 1.3 introduit une première manière de sélectionner les paramétrages souhaités totalement calquée sur la logique de Hoare (1969): à partir d'une trace observée expérimentalement et transcrite dans un langage formel, il est possible de construire les contraintes qui rendent cette trace possible dans le modèle formel (Bernot *et al.* (2018)).

Une deuxième approche, basée sur le Model-Checking est présentée en section 1.4: cette fois-ci on transcrit en formule d'une logique temporelle les propriétés connues du système biologique. Le but de jeu est alors de calculer *exhaustivement* l'ensemble des paramétrages qui mènent à des dynamiques compatibles avec ces propriétés. Généralement, même avec des observations partielles, il reste possible d'écrire une formule temporelle représentant ce que l'on a observé (Bernot *et al.* (2004)).

La section 1.5 présente la plate-forme *TotemBioNet* dédiée à l'identification des paramètres pour la formalisation discrète de René Thomas en exploitant la complémentarité des deux approches basées sur la logique de Hoare et la logique temporelle.

Par ailleurs, le cadre de R. Thomas, ayant fait le choix d'abstraire le temps chronométrique, n'est pas capable de décrire plus que les successions chronologiques d'événements. Lorsqu'on souhaite modéliser un système biologique dont la composante chronométrique est importante, il faut alors étendre le cadre de modélisation. En section 1.6, nous présentons un cadre hybride qui reprend une bonne partie de la modélisation qualitative de R. Thomas et qui la complète afin d'être en mesure de décrire quantitativement le temps passé dans chacun des états qualitatifs.

Pour faire face au problème de l'identification des paramètres, nous introduisons ensuite en section 1.7 une logique de Hoare hybride afin de construire les contraintes sur les paramètres cinétiques du modèle qui sont suffisantes et nécessaires pour que le modèle exhibe une trace chronométrique donnée. Enfin, la section 1.8 propose en conclusion une méthodologie assistée par les méthodes et outils présentés dans ce chapitre pour développer des modèles pertinents.

Exemple illustratif : le cycle circadien simplifié des mammifères

Chez les mammifères, de très nombreux rythmes biologiques ont une périodicité de 24 heures. Ce phénomène traduit la coordination temporelle des processus biologiques pour adapter l'organisme aux changements de l'environnement qui ont lieu au cours de la journée, et d'optimiser la physiologie de l'organisme en permettant au métabolisme d'anticiper les variations quotidiennes de l'environnement. Le système qui permet ces anticipations, dont l'adaptation du vivant à l'alternance jour/nuit, est appelé *horloge circadienne* (du grec *circa*, autour et *dies*, jour). Parmi les adaptations classiques, l'horloge contrôle la température, la

réparation des tissus, le sommeil, *etc.* Cette horloge compte parmi les mécanismes physiologiques les plus importants (43% des gènes sont régulés par ce système chez la souris, voir Zhang *et al.* (2014)). Au-delà des mammifères, l'immense majorité des êtres vivants étant soumis à l'oscillation jour/nuit ou à ses conséquences, l'horloge circadienne est omniprésente dans le vivant.

Dans tout le reste de ce chapitre, nous noterons *HCSM*, pour *Horloge Circadienne Simplifiée des Mammifères*, le modèle que nous décrivons intuitivement ci-dessous.

Le réseau de régulation qui réalise la fonction principale de l'horloge circadienne dans le noyau suprachiasmatique peut être vu, de manière très abstraite, comme un jeu d'influences entre un ensemble de gènes (dits *gènes de l'horloge*) et un ensemble de protéines dont les principales (*Per* et *Cry*) forment un complexe (*Per-Cry*). Plusieurs facteurs extérieurs contrôlent partiellement ce jeu d'influences, le facteur principal étant l'alternance jour/nuit mais d'autres facteurs comme la prise alimentaire ou l'activité physique entrent en jeu également.

Dans les très grandes lignes, les gènes de l'horloge, notés *G* dans la figure 1.1, produisent les protéines *Per* et *Cry*, et par conséquent les complexes *Per-Cry*, notés *PC* dans la figure. On peut donc considérer que *G* est un *activateur* de *PC* (flèche annotée par un +). Réciproquement, les complexes *Per-Cry* peuvent retourner dans

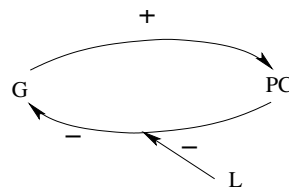


Figure 1.1. *Horloge circadienne simplifiée des mammifères: les gènes de l'horloge *G* produisent les protéines *Per-Cry*, *PC*, qui, à leur tour, inhibent l'expression de *G*. Mais la lumière *L* inhibe cette inhibition.*

le noyau de la cellule après leur production et y jouent alors un rôle *inhibiteur* des gènes de l'horloge (flèche annotée par – de *PC* à *G*). Cette vision très simplifiée a l'avantage de faire comprendre aisément le phénomène circadien oscillatoire: si *G* s'exprime, il fait s'exprimer *PC*, qui en retour empêche *G* de continuer à s'exprimer, donc *G* ne produit plus *PC* qui est donc dégradé, dès lors *G* n'est plus inhibé et recommence à s'exprimer, *etc.*

Cette oscillation dure *environ* 24h, et c'est en fait la présence ou absence de la lumière qui va contrôler plus finement sa durée: en présence de lumière (*L*)

l'inhibition de PC est rendue inopérante et l'expression de G favorisée. On peut donc considérer que la lumière inhibe l'inhibition de G par PC .

1.2. Formalisation de la modélisation de R. Thomas

La modélisation logique définie dans Thomas (1991a) pour décrire et analyser les réseaux de régulation biologique, voir aussi Thomas and d'Ari (1990), s'appuie sur une représentation du système en terme de *graphes*. La description *statique* du système est définie par un *graphe d'influences*, tandis que le comportement *dynamique* du système est étudié grâce à la notion de *graphe de transitions*. Nous présentons de façon informelle le lien entre composant d'un graphe et notion biologique modélisée, pour construire briques après briques les définitions sur lesquelles s'appuient les méthodes formelles que nous préconisons.

1.2.1. Description statique ou graphe d'influences

1.2.1.1. Des sommets et des arcs pour décrire l'influence entre les gènes

De façon informelle, un graphe de régulation ou *graphe d'influences*, représente les influences individuelles de chaque gène sur les autres gènes (ou toute autre entité biologique étudiée).

Les sommets du *graphe d'influences* sont les gènes, ou de façon plus générale les *variables*. Un *arc* entre v_1 et v_2 signifie que v_1 a une influence sur v_2 : dans un certain état, c'est à dire quand la valeur (du produit actif) de v_1 a franchi un certain *seuil*, v_1 a une influence sur l'état de v_2 .

1.2.1.2. Arcs sortants et seuils d'activation

Quand une variable v possède n arcs sortants, elle peut influencer n autres variables $v_1, v_2, \dots, v_n$. Pour tout successeur v_i , un seuil est associé à l'action de v sur v_i : si v franchit ce seuil, alors l'influence de v sur v_i peut avoir lieu. S'il s'agit d'une activation, v_i bénéficie de l'aide de v pour s'exprimer¹ alors que s'il s'agit d'une inhibition, v_i a tendance à diminuer son expression. Une variable v a donc au plus $n + 1$ valeurs: la valeur 0 est le cas où v n'a aucune influence sur ses successeurs, et pour tout i , la valeur i correspond au cas où les i plus petits seuils ont été franchis. Comme il est possible qu'un même seuil de la variable v permette d'influencer plusieurs de ses successeurs, le nombre de seuils de v peut être inférieur au nombre de ses successeurs.

1. Ces seuils correspondent à la discrétisation des fonctions sigmoïdes de la théorie de R. Thomas.

NOTATION 1.1.– Le domaine d'une variable v est le domaine discret noté $\llbracket 0, b_v \rrbracket$ ² avec $b_v \leq d^+(v)$ où $d^+(v)$ est le degré sortant de v dans le graphe d'influences.

Notons un point important: les seuils fournissent des informations *qualitatives*. Il s'agit d'exprimer que le niveau de v est suffisant pour influencer v_i mais il ne s'agit pas d'une information *quantitative*, le seuil n'est donc pas une concentration réelle.

EXAMPLE 1.1.– HCSM. En figure 1.2, chaque variable ayant un unique successeur, les domaines de G , L et PC sont $\llbracket 0, 1 \rrbracket$.

1.2.1.3. Étiquette des arcs et conditions d'activation ou inhibition

L'influence d'un gène sur un autre peut être une *inhibition* ou une *activation* qui n'est effective que passé un certain seuil. Les arcs doivent donc être étiquetés par ces deux informations (signe de l'influence et seuil). Nous proposons de les représenter par un atome de la logique propositionnelle.

NOTATION 1.2.– Un arc de v_1 vers v_2 est étiqueté par $(v_1 \geq s)$ pour indiquer que v_1 est un activateur de v_2 passé le seuil s , l'arc est étiqueté par $\neg(v_1 \geq s)$ si v_1 est un inhibiteur de v_2 passé le seuil s .

DEFINITION 1.1 (Graphe d'influences étiqueté).– Un graphe d'influences étiqueté est défini par $\Sigma_e = (V, A)$ avec:

- V , l'ensemble des sommets de domaines discrets $\llbracket 0, b_v \rrbracket$ où b_v est la borne associée à la variable v , et
- A , l'ensemble des arcs entre les sommets de V , $v \rightarrow v'$, étiquetés par des formules atomiques de la forme $(v \geq s)$ ou bien $\neg(v \geq s)$ où $s \in \llbracket 0, b_v \rrbracket$.

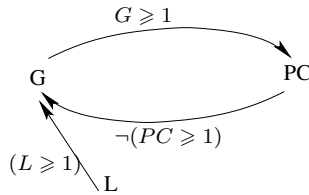


Figure 1.2. Graphe d'influences du HCSM

1.2.1.4. Arcs incidents et paramètres

Dans le graphe d'influences, alors que les arcs sortants de v désignent les variables cibles de v , les arcs incidents sur v spécifient les variables susceptibles d'influencer v .

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

Quand une variable v du graphe d'influences a plusieurs *arcs incidents* elle peut subir l'influence d'un de ses prédécesseurs de façon individuelle ou d'une combinaison de ses prédécesseurs. La notion de *paramètres*, que nous noterons K_v pour la variable v , aide à désigner ces combinaisons d'influences. Ainsi, si v a n prédécesseurs $p_1, p_2, \dots, p_n$, les paramètres associés à v sont $K_{v,\{\}} qui représente le fait que v n'est influencé par aucun prédécesseur, $K_{v,\{p_i\}}$ pour indiquer que v est influencé par p_i , $K_{v,\{p_1,p_2\}}$ quand v est influencé à la fois par p_1 et p_2 , et de même pour toute combinaison possible entre les p_i . Il s'en suit que si v a n arcs incidents, on lui associe $2^{d^-(v)}$ paramètres où $d^-(v)$ est le degré entrant de v .$

NOTATION 1.3.– Les paramètres associés à la variable v sont notés $K_{v,\omega}$ où ω est un sous-ensemble des prédécesseurs de v (i.e. $\omega \subseteq \Sigma^{-1}(v)$ avec $\Sigma^{-1}(v)$ l'ensemble des prédécesseurs de v dans le graphe d'influences).

EXAMPLE 1.2.– HCSM. Pour la figure 1.2, les paramètres associés à G sont $K_{G,\{\}}, K_{G,\{PC\}}, K_{G,\{L\}}, K_{G,\{PC,L\}}$. L a un seul paramètre $K_{L,\{\}}$. Les paramètres associés à PC sont $K_{PC,\{\}}$ et $K_{PC,\{G\}}$.

1.2.2. Dynamique d'un graphe de régulation biologique

Une fois le graphe d'influences défini, le travail du modélisateur est de fournir une aide au biologiste pour élucider le comportement dynamique du système. Dans la modélisation de R. Thomas, la dynamique du système est représentée par un *graphe de transitions* qui permet partant d'un état quelconque du système de fournir les séquences d'états qui peuvent se produire: “In the graph of sequences of states, the vertices represent the logical states of the system and the edges (arrows) represent the transitions between states”, Thomas and d'Ari (1990).

Usuellement, l'ensemble des états du système est le produit cartésien du domaine des variables. Dans le cadre de la logique formelle que nous utilisons pour analyser la dynamique du système, un état est vu comme une *interprétation* des variables dans $\llbracket 0, b_v \rrbracket$.

DEFINITION 1.2 (Etat).– Un état η d'un graphe d'influences étiqueté est une interprétation qui associe à chaque variable v de V une valeur dans $\llbracket 0, b_v \rrbracket$. On note ζ l'ensemble des états du système, et η_v la valeur de v pour l'interprétation η .

Pour analyser le *comportement dynamique* du système, deux questions doivent être abordées:

Q1 Comment caractériser un changement d'état ?

Q2 Quelle est la nouvelle valeur d'une variable qui a changé d'état ?

Nous expliquons tout d'abord le rôle fondamental des *paramètres* pour répondre à ces deux questions, puis nous définissons de façon formelle la dynamique du système.

1.2.2.1. Paramètres pour désigner les ressources

Pour un état donné $\eta \in \zeta$, et une variable v , un changement d'état de v peut se produire de trois façons:

- 1) si un activateur de v dans le graphe d'influences vient d'atteindre le seuil d'influence qui aide v à s'exprimer, ou bien si un inhibiteur de v vient de passer en dessous du seuil suffisant pour inhiber v , alors il se peut que cette « ressource » supplémentaire permette à v d'augmenter,
- 2) si au contraire un activateur de v dans le graphe d'influences vient de passer en dessous du seuil suffisant pour aider v à s'exprimer, ou bien si un inhibiteur de v vient d'atteindre le seuil suffisant pour inhiber v , alors il se peut que cette perte de ressource permette à v de diminuer,
- 3) enfin, même lorsque aucun prédécesseur n'a passé son seuil suffisant pour réguler v (dans un sens ou l'autre), la valeur de v peut encore évoluer dans le cas où elle n'est pas encore stabilisée.

Un prédécesseur de v qui a atteint dans l'état η le seuil d'activation de v est appelé *ressource* de v dans l'état η . Il en est de même d'un inhibiteur qui n'atteint pas le seuil d'inhibition de v . Ainsi en un état η donné, pour chaque variable v , on peut calculer l'unique ensemble ω des *ressources* de v , et comme le paramètre associé $K_{v,\omega}$ représente l'effet combiné des influences de chacune des ressources $p \in \omega$, il n'existe qu'un seul paramètre $K_{v,\omega}$ où ω est l'ensemble des ressources en η . Nous dirons que ce paramètre est *applicable*.

DEFINITION 1.3 (Paramètre applicable).— Soit $\eta \in \zeta$ un état d'un graphe d'influences (V, A) . Soit v une variable dans V . Le paramètre applicable de v sur l'état η est le paramètre $K_{v,\omega}$ où ω est l'ensemble des variables p_i telles que les formules associées aux arcs (p_i, v) , $p_i \in \omega$ sont vraies dans l'état η^3 .

REMARQUE 1.4.— Si aucune de ces formules n'est vraie, v n'a aucun prédécesseur ressource sur η et le paramètre applicable est $K_{v,\{\}}$.

Ceci répond à la question Q1: un changement d'état d'une variable v va être caractérisé par le paramètre $K_{v,\omega}$ applicable sur l'état η .

EXEMPLE 1.3.— *HCSM*. Soit l'état $\eta = (G, L, PC) = (1, 1, 0)$. Alors le paramètre applicable de PC est $K_{PC,\{G\}}$ puisque le seul prédécesseur de PC est G , et que l'arc (G, PC) est étiqueté par $(G \geq 1)$ qui est vraie pour η . Le paramètre applicable

3. Vrai s'entend ici avec l'interprétation η pour les variables et le sens usuel de $\geq$ dans $\mathbb{N}$.

de G est $K_{G,\{L,PC\}}$. En effet, les deux prédécesseurs de G sont PC et L . La formule associée à l'arc (L, G) est $(L \geq 1)$, celle associée à l'arc (PC, G) est $\neg(PC \geq 1)$, et ces deux formules sont vraies pour η . Enfin, le paramètre applicable de L est $K_{L,\{\}}$ puisque l'ensemble des prédécesseurs de L est vide.

1.2.2.2. Valeurs des paramètres comme nouvelles valeurs des variables

La question Q2 consiste à déterminer la valeur d'une variable v après transition d'état. L'interprétation η de v est à valeur dans $\llbracket 0, b_v \rrbracket$ et un changement d'état de v est caractérisé par le paramètre $K_{v,\omega}$ applicable pour l'état η . Si l'on considère toutes les interprétations de $K_{v,\omega}$ dans $\llbracket 0, b_v \rrbracket$, une transition d'état possible est le passage de la valeur courante de η_v à la valeur $K_{v,\omega}$. Ceci répond en partie à la question Q2: un prochain état possible pour v serait la valeur de son paramètre applicable pour une interprétation dans $\llbracket 0, b_v \rrbracket$ (on verra en section 1.2.2.4 que c'est légèrement plus subtil).

Notons que le terme de *paramètre* (cf section 1.2.1.4) choisi pour désigner les combinaisons d'influences des arcs incidents sur v prend ici tout son sens: il s'agit bien de *paramètres* puisque la dynamique du système dépend de la valeur des K_v . Il peut *a priori* y avoir autant de dynamiques du système que d'interprétations des K_v dans $\llbracket 0, b_v \rrbracket$, et ce pour toute variable v , c'est à dire au pire $\prod_v (d^+(v) + 1)^{2^{d^-(v)}}$ dynamiques.

EXEMPLE 1.4. – *HCSM*. Le nombre total de dynamiques est $2^{2^1} \times 2^{2^0} \times 2^{2^2} = 128$ puisque les trois variables sont à domaine $\llbracket 0, 1 \rrbracket$, que G a un prédécesseur, L n'a pas de prédécesseurs et PC a deux prédécesseurs. Si l'on considère l'état $(G, L, PC) = (1, 1, 0)$ et les interprétations des paramètres applicables: $K_{G,\{L,PC\}} = 0$, $K_{L,\{\}} = 0$ et $K_{PC,\{G\}} = 1$, alors le nouvel état du système est $(G, L, PC) = (0, 0, 1)$. Ce choix de paramètres fait se coucher le jour puisque L passe de 1 à 0.

1.2.2.3. Multiplexes pour réduire le nombre de dynamiques

Dans certains cas, les connaissances des biologistes permettent de réduire le nombre de paramètres, et par conséquent le nombre de dynamiques qu'il est possible d'associer au graphe d'influences. Par exemple, lorsque deux protéines forment un complexe avant d'agir sur un gène cible, plutôt que de considérer les deux arcs (un arc partant de chacune des protéines vers la cible commune), on peut considérer un unique arc étiqueté par un *multiplexe* jouant le rôle du complexe. La condition d'applicabilité de ce *multiplexe* est alors la conjonction des conditions d'action de chacune des protéines.

La notion de *multiplexe* introduite dans Khalis, Bernot and Comet (2009) généralise la notion d'étiquette des arcs présentée en 1.1, par un couple formé d'un identificateur m et d'une formule φ_m .

DEFINITION 1.4 (Langage des formules de multiplexes de $(V \cup M)$).— Soit V un ensemble de variables v de domaines discrets $\llbracket 0, b_v \rrbracket$, soit M un ensemble d'identificateurs de multiplexes. Le langage des formules de multiplexes de $(V \cup M)$ est défini inductivement par:

- les atomes sont les identifiants de M et les formules atomiques de la forme $(v \geq n)$ avec $v \in V$ et $n \in \llbracket 0, b_v \rrbracket$,
- les formules sont de la forme $\neg\varphi$, $\varphi_1 \square \varphi_2$ où φ , φ_1 et φ_2 sont des formules de multiplexes et $\square$ est $\wedge$, $\vee$ ou $\Rightarrow$.

DEFINITION 1.5 (Graphe d'influences avec multiplexes).— Un graphe d'influences avec multiplexes est défini par $\Sigma = (V \cup M, A)$ où $V \cup M$ est l'ensemble des sommets et A est l'ensemble des arcs, avec:

- V , l'ensemble des variables v de domaines discrets $\llbracket 0, b_v \rrbracket$,
- M , l'ensemble des multiplexes m de formules associées φ_m , où les φ_m sont des formules de multiplexes de $(V \cup M)$
- $A \subset (M \times V)$.

REMARQUE 1.5.— Dans la définition 1.1 (graphes d'influences étiquetés), les arcs relient les variables et sont étiquetés par des formules atomiques. Ici, les arcs vont des multiplexes vers les variables. On pourrait rajouter des arcs allant des variables vers les multiplexes (pour retrouver les influences entre variables), mais ces arcs de $(V \times M)$ seraient redondants puisque la formule associée à chaque multiplexe indique les prédécesseurs jouant un rôle de régulateurs. Par exemple, la formule $\neg(v_1 \geq 2) \wedge (v_1 \geq 1)$ associée au multiplexe m indique clairement que v_1 et v_2 sont les prédécesseurs de m .

EXEMPLE 1.5.— HCSM. Une autre façon d'exprimer l'influence combinée de PC et L sur G est d'utiliser un multiplexe de formule associée $\neg(PC \geq 1 \wedge \neg(L \geq 1))$. On obtient alors le graphe d'influences avec multiplexes de la figure 1.3.

1.2.2.4. Définition formelle de la dynamique d'un système

Comme expliqué en 1.2.2.1 et 1.2.2.2, les paramètres jouent un rôle fondamental dans la dynamique du graphe d'influences. Nous nommons *réseau de régulation* le couple formé du graphe et des paramètres qui lui sont associés.

DEFINITION 1.6 (Réseau de régulation).— Un réseau de régulation $(\Sigma, \mathcal{K})$ est la donnée d'un graphe d'influences avec multiplexes Σ auquel on associe la famille de paramètres $\mathcal{K} = \{K_{v,\omega} | v \in V \text{ et } \omega \subseteq \Sigma^{-1}(v)\}$ où $\Sigma^{-1}(v)$ est l'ensemble des prédécesseurs de v dans le graphe d'influences avec multiplexes.

REMARQUE 1.6.— Dans un graphe d'influences avec multiplexes, les prédécesseurs des variables sont des multiplexes. Les paramètres K sont donc de la forme $K_{v,\omega}$ où

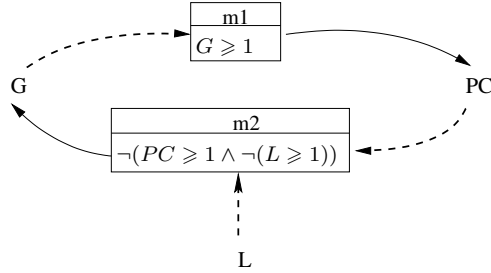


Figure 1.3. Graphe d'influences avec multiplexes du HCSM.
Les flèches en pointillés n'apparaissent pas dans la définition du graphe mais aident à sa visualisation.

$\omega \subseteq M$. Le paramètre applicable pour v est celui tel que ω est exactement l'ensemble des multiplexes dont les formules associées sont vraies dans l'état courant (donc les formules des autres prédécesseurs de v sont fausses).

EXAMPLE 1.6. – HCSM. Le réseau de régulation biologique est le couple formé du graphe de la figure 1.3 et de la famille $\mathcal{K}$ constituée de: $K_{G,\{\}}, K_{G,\{m2\}}, K_{L,\{)}, K_{PC,\{\}} et $K_{PC,\{m1\}}$. Le nombre total de dynamiques potentielles de ce réseau est $2^{2^1} \times 2^{2^0} \times 2^{2^1} = 32$. Le nombre de dynamiques potentielles a donc été divisé par 4 grâce à l'introduction des multiplexes.$

La notion de *graphe de transitions* définit les transitions asynchrones possibles d'un réseau de régulation.

DEFINITION 1.7 (graphe de transitions). – Un graphe de transitions associé au réseau de régulation $(\Sigma, \mathcal{K})$ est défini de la façon suivante:

- L'ensemble des sommets est l'ensemble des états possibles ζ
- Les transitions entre un état $\eta = (\eta_{v_1}, \eta_{v_2}, \dots, \eta_{v_n})$ et l'état suivant $\eta' = (\eta'_{v_1}, \eta'_{v_2}, \dots, \eta'_{v_n})$ sont celles telles que:
 - 1) Il existe un unique i tel que $\eta_{v_i} \neq \eta'_{v_i}$,
 - 2) Soit v_i tel que $\eta_{v_i} \neq \eta'_{v_i}$, et soit $K_{v_i,\omega}$ le paramètre applicable de v_i sur η . On a $K_{v_i,\omega} \neq \eta_{v_i}$ et: si $K_{v_i,\omega} > \eta_{v_i}$ alors $\eta'_{v_i} = \eta_{v_i} + 1$, si $K_{v_i,\omega} < \eta_{v_i}$ alors $\eta'_{v_i} = \eta_{v_i} - 1$.

Le point 1 exprime qu'une seule variable change d'état à chaque transition, on dit que le graphe de transitions est *asynchrone*. Dans le cas où plusieurs variables n'ont pas atteint la valeur de leur paramètre applicable dans l'état courant, une seule est sélectionnée de façon non déterministe pour effectuer la transition d'état (mais il y a

autant de transitions que de telles variables). Le point 2 exprime que les changements se font par paliers de 1 en se rapprochant de la valeur du paramètre applicable, parfois appelée valeur focale.

Puisque la valeur des variables dépend de la valeur des K , il n'y a pas un unique *graphe de transitions* pour un réseau de régulation donné, mais autant de graphes que d'interprétations correctes des $\mathcal{K}$.

DEFINITION 1.8 (Interprétation correcte de $(\Sigma, \mathcal{K})$).– Une interprétation correcte de la famille des $\mathcal{K}$, notée $\sigma : \mathcal{K} \rightarrow \mathbb{N}$, associe à chacun des $K_{v,\omega}$ une valeur dans $\llbracket 0, b_v \rrbracket$. On note $\mathcal{S}_\Sigma$ l'ensemble des interprétations correctes de $(\Sigma, \mathcal{K})$.

DEFINITION 1.9 (Dynamique d'un réseau de régulation).– Soit $(\Sigma, \mathcal{K})$ un réseau de régulation. Soit $\mathcal{T}$ l'ensemble des graphes de transitions associés à $(\Sigma, \mathcal{K})$. Soit $\mathcal{S}_\Sigma$ l'ensemble des interprétations correctes de $\mathcal{K}$. La dynamique de $(\Sigma, \mathcal{K})$ est une fonction $\mathcal{D}$ qui associe à chaque interprétation correcte des $\mathcal{K}$ le graphe de transitions de la définition 1.7:

$$\begin{array}{ccc} \mathcal{D} : \mathcal{S}_\Sigma & \rightarrow & \mathcal{T} \\ \sigma & \mapsto & \tau \end{array}$$

EXEMPLE 1.7.– *HCSM*. On s'intéresse à la dynamique du système pendant la nuit donc lorsque $L = 0$. Les paramètres applicables en fonction des états sont les suivants:

G	PC	K_G	K_{PC}
0	0	$K_{G,\{m_2\}}$	$K_{PC,\{\}} \}$
0	1	$K_{G,\{\}}$	$K_{PC,\{\}}$
1	0	$K_{G,\{m_2\}}$	$K_{PC,\{m_1\}}$
1	1	$K_{G,\{\}}$	$K_{PC,\{m_1\}}$

La figure 1.4 présente à gauche le graphe obtenu avec l'interprétation $\sigma_1 : (K_{G,\{\}}, K_{G,\{m_2\}}, K_{PC,\{\}}, K_{PC,\{m_1\}}) \mapsto (0, 1, 0, 1)$ et à droite le graphe obtenu pour $\sigma_2 : (K_{G,\{\}}, K_{G,\{m_2\}}, K_{PC,\{\}}, K_{PC,\{m_1\}}) \mapsto (0, 0, 0, 0)$.

1.2.2.5. Identification des paramètres

Etudier la dynamique du système revient à étudier l'ensemble $\mathcal{T}$ des graphes de transitions. Cet ensemble étant très grand, toute la difficulté réside dans l'identification d'un sous-ensemble de $\mathcal{T}$ dont les graphes de transitions soient cohérents avec les propriétés biologiques en cours d'élaboration. De plus, il est indispensable de réduire au maximum ce sous-ensemble pour qu'il soit analysable par l'humain afin de pouvoir mettre en place un protocole expérimental pour vérifier des hypothèses biologiques. Trouver ce sous-ensemble de $\mathcal{T}$ revient à trouver un sous-ensemble de $\mathcal{S}_\Sigma$, en d'autres termes, il s'agit bien d'identifier la valeur des paramètres K .

Dans certains cas, les connaissances biologiques permettent de fixer directement la valeur de certains paramètres, ou tout au moins de réduire les valeurs possibles

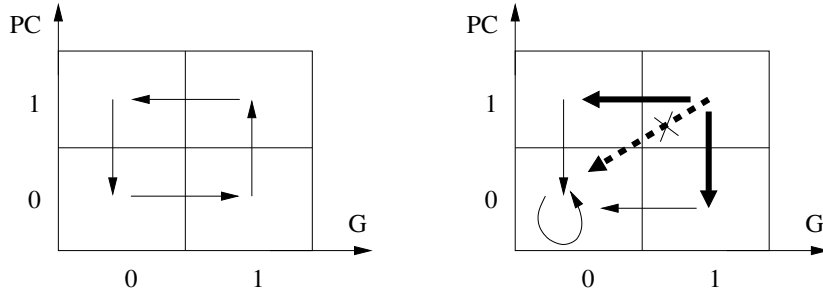


Figure 1.4. Deux graphes de transitions du HCSM de nuit.

A gauche, un graphe qui reflète l'oscillation circadienne, et à droite un graphe moins crédible biologiquement mais qui montre bien le principe de désynchronisation: dans l'état $(1,1)$ la flèche en pointillée qui modifierait les deux variables en même temps est remplacée par les deux transitions en gras qui ne modifient que l'une des deux à la fois.

pour ce paramètre. Nous illustrons ce propos en complétant la dynamique de l'horloge circadienne simplifiée. En effet, telle que présenté jusqu'à présent, la lumière est présente ou absente mais ne joue pas son rôle de contrôleur de l'horloge, ou "Zeitgeber"⁴. Pour cela, il faut que L oscille entre les valeurs 0 et 1. Une solution est d'ajouter une auto-régulation de L (Cf. figure 1.5) qui est désormais gérée par les paramètres $K_{L,\{\}} et $K_{L,\{m_3\}}$. Seules les interprétations où $K_{L,\{\}} \neq K_{L,\{m_3\}}$, comme par exemple $\sigma(K_{L,\{\}}) = 0$ et $\sigma(K_{L,\{m_3\}}) = 1$, permettront l'oscillation.$

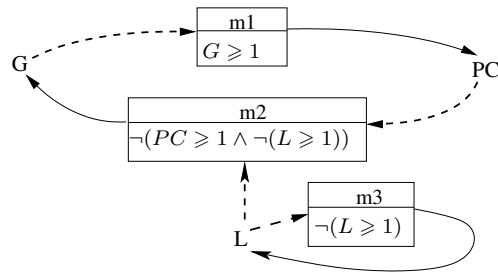


Figure 1.5. Graphe d'influences avec multiplexes et Zeitgeber du HCSM.

4. Littéralement, *donneur de temps*.

Dans le cas général, l'identification des paramètres est un problème difficile. Il faudra combiner:

- l'expertise du biologiste pour extraire des informations sur la dynamique du système, par exemple, des propriétés exprimant qu'un état stable sera atteint sous certaines conditions, ou bien, l'observation d'une trace expérimentale permettant de décrire les changements d'états de certains gènes,
- et les méthodes de vérification formelle qui pourront exploiter ces informations et réduire ainsi l'espace de recherche.

C'est ce dernier point que nous développons dans les prochaines sections.

1.3. Logique de Hoare génétiquement modifiée

En pratique, le graphe d'influences avec multiplexes est obtenu à l'issue des premières discussions avec les partenaires biologistes, à partir de la bibliographie scientifique et en accord avec le focus (*i.e.* la question de recherche biologique) qui les motive. Même s'il arrive de modifier ce graphe *statique* au cours de la modélisation, le problème majeur de la modélisation reste l'identification des paramètres, c'est-à-dire la recherche du sous-ensemble de $\mathcal{S}_\Sigma$ contenant les interprétations compatibles avec les connaissances biologiques sur la *dynamique* du système (graphes de transitions compatibles).

1.3.1. Exploitation des observations expérimentales, sur un exemple

Outre les connaissances acquises que l'on peut formaliser en logique temporelle (section 1.4), des courbes de niveaux d'expression obtenues avec une bonne résolution temporelle, et après élimination du bruit, sont un moyen performant de réduire l'espace de recherche. Imaginons par exemple un protocole expérimental où l'on place le sujet biologique en pleine lumière ($L = 1$) de 8h à 20h et en conditions nocturnes ($L = 0$) de 20h à 8h. Un lot d'expériences, où l'on mesure à la fois la production des gènes de l'horloge G et la présence du complexe Per-Cry (PC) dans la cellule, peut aboutir à des courbes moyennes similaires à celles de la figure 1.6.

On constate alors qu'entre 8h et 10h, la lumière est allumée ($L = 1$), les gènes de l'horloge ne s'expriment pas de manière notable ($G = 0$), et les complexes Per-Cry sont dégradés (transition $PC = 1$ à $PC = 0$). De plus, d'après le graphe d'influences de la figure 1.5, la formule du multiplexe m_1 est non satisfaite puisque $G = 0$, donc le paramètre applicable pour PC est $K_{PC, \{ \}}$. Si la valeur de ce paramètre était égale à 1, on ne pourrait pas observer la transition de $PC = 1$ à $PC = 0$ (dernier alinéa de la définition 1.7): il est nécessaire que $\sigma(K_{PC, \{ \}})$ soit strictement inférieure à la valeur courante de PC pour que PC diminue. Donc $\sigma(K_{PC, \{ \}}) = 0$.

L'investigation des conséquences de ces courbes expérimentales peut continuer:

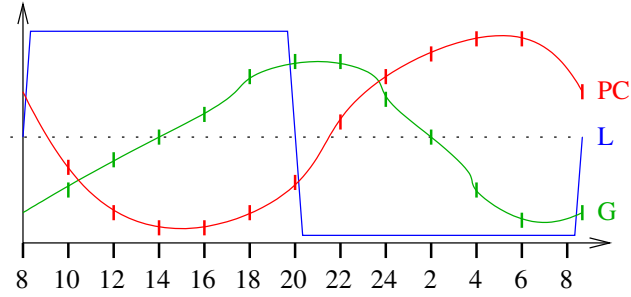


Figure 1.6. Exemple de courbes expérimentales: après allumage de la lumière L , Per-Cry (PC) est dégradé, puis les gènes de l'horloge G s'expriment, puis la lumière est éteinte, les complexes Per-Cry sont produits par les gènes de l'horloge et les gènes de l'horloge sont inhibés.

– entre 10h et 20h, la lumière reste allumée ($L = 1$), les complexes Per-Cry restent peu présents ($PC = 0$), et les gènes de l'horloge commencent à s'exprimer de manière notable (transition $G = 0$ à $G = 1$). Toujours d'après le graphe d'influences de la figure 1.5, la formule du multiplexe m_2 étant satisfaite (car $0 \geq 1 \wedge \neg(1 \geq 1)$ est faux), on en déduit que $K_{G, \{m_2\}}$ est le paramètre applicable de G , donc $\sigma(K_{G, \{m_2\}})$ est strictement supérieure à la valeur courante de G , donc $\sigma(K_{G, \{m_2\}}) = 1$

– à 20h, $PC = 0$, $G = 1$ et la lumière s'éteint (transition $L = 1$ vers $L = 0$). Juste avant l'extinction, la formule du multiplexe m_3 était fausse, par conséquent le paramètre applicable est $K_{L, \{\}}$, donc $\sigma(K_{L, \{\}})$ est strictement inférieure à la valeur courante de L , donc $\sigma(K_{L, \{\}}) = 0$

– de même entre 20h et 22h PC passe de 0 à 1 alors que m_1 est ressource (i.e. $(G \geq 1)$ est satisfaite), donc $\sigma(K_{PC, \{m_1\}}) = 1$

– entre 22h et 8h G passe de 1 à 0 alors que $\neg(0 \geq 1 \wedge \neg(1 \geq 1))$ est fausse, donc $\sigma(K_{G, \{\}}) = 0$

– enfin la lumière s'allume à 8h, donc $\sigma(K_{L, \{m_3\}}) = 1$.

Ainsi les observations expérimentales de la figure 1.6 suffisent à déterminer la valeur de tous les paramètres.

Ce n'est pas toujours aussi facile. La figure 1.6 repose sur un cas idéal où toutes les variables sont booléennes et en faible nombre, où l'on a su calibrer les courbes pour identifier facilement un seuil entre 0 et 1, et où l'état initial de toutes les variables est connu. Sur des questions biologiques plus complexes, il peut arriver que les variables ne soient pas booléennes, que les seuils ne soient connus qu'approximativement, et que l'état initial (resp. final) de l'expérience ne soit connu que partiellement. Ainsi

certaines ambiguïtés peuvent apparaître, notamment sur l'ordre de modification des variables (et la mise à jour asynchrone de la théorie de Thomas impose qu'une seule variable à la fois change de valeur).

La démarche décrite *via* cet exemple n'est pas sans rappeler les techniques d'analyse statique de logiciels et d'interprétation abstraite, au nombre desquelles la *logique de Hoare* (1969) est un grand classique. Nous allons voir comment généraliser notre démarche avec une logique de Hoare « génétiquement modifiée » (Bernot *et al.* (2018)).

1.3.2. Un langage d'assertions

Tout d'abord, si l'on ne connaît pas totalement les états initiaux et finaux des courbes expérimentales (par une précision insuffisante des mesures ou par méconnaissance de la place exacte des seuils), on dispose néanmoins d'indications précieuses. Le langage des multiplexes (définition 1.4) pourrait suffire à décrire les plages d'états initiaux ou finaux possibles sous des formes telles que $(v \geq 1 \wedge \neg(v \geq 3) \wedge v' \geq 2 \wedge \dots)$. Toutefois, on peut également vouloir importer des connaissances issues d'autres expériences ou de la littérature biologique, telles que $(K_{v,\{x_1,x_2\}} \geq K_{v,\{\}} + 2)$. On s'autorise donc un langage d'assertions étendu :

DEFINITION 1.10 (Langage d'assertions).— Soit $(\Sigma = (V \cup M, A), \mathcal{K})$ un réseau de régulation biologique. Le langage d'assertion sur ce réseau est défini inductivement par :

- les atomes sont les expressions de la forme $(t \geq t')$ où t et t' sont des termes bien formés à partir des symboles de V , de $\mathcal{K}$, de $\mathbb{N}$ et des opérations binaires $+$ et $-$.
- si φ_1 et φ_2 sont des assertions alors $\neg\varphi$ et $\varphi_1 \wedge \varphi_2$ sont également des assertions.

Les connaissances préalables aux mesures faites peuvent alors être écrites sous la forme d'une assertion, appelée la *pré-condition*. Et les connaissances obtenues à l'issue du déroulement expérimental, peuvent être écrites sous la forme d'une assertion, appelée la *post-condition*.

1.3.3. Un langage des chemins

Par rapport à un programme impératif classique en informatique, les transitions de variables sont beaucoup plus simples, puisqu'on ne peut qu'incrémenter ou décrémenter de 1 la valeur d'une variable. Un changement élémentaire le long de nos suivis de courbes peut donc s'écrire simplement $v+$ si la variable v augmente, ou $v-$ si elle diminue. Par ailleurs, des observations biologiques peuvent être effectuées *au cours* des expériences : on peut les exprimer par des assertions placées au beau milieu des observations de la forme $v+$ ou $v-$. Par exemple, suivant les courbes de la

figure 1.6, supposons que l'on ait observé vers 22h que les gènes de l'horloge atteignent leur maximum: $G = 1$ à ce moment là car sinon le seuil nécessaire pour produire PC serait supérieur à l'expression maximale des gènes de l'horloge qui ne produiraient donc jamais de complexe Per-Cry. On pourrait abstraire ces mesures et observations successives par: $PC-; G+; L-; PC+; (G = 1); G-; L+$. De plus, si l'on effectue un Knock-Out (resp. une saturation) d'une variable v en cours d'expérience, on peut écrire $v:=0$ (resp. $v:=b_v$), qui affecte une nouvelle valeur à la variable v .

Il peut arriver que deux passages de seuils s'effectuent presque en même temps, ou bien que les mesures ne soient pas assez précises pour savoir quelle variable passe son seuil en premier, ou bien qu'on observe plusieurs changements d'état à un point de mesure, etc. Lorsque de telles incertitudes apparaissent, plusieurs chemins peuvent traduire ce qui s'est passé, qui diffèrent au moins par l'ordre des observations. On sait que l'un au moins de ces chemins a été emprunté lors des expériences mais on ne peut déterminer lequel. On indique une telle incertitude par une construction existentielle. Par exemple, si c'est à 21h qu'on a observé que les gènes de l'horloge atteignent leur maximum, cette observation peut aussi bien prendre place avant ou après que PC passe son seuil, sans réel moyen d'en déterminer l'ordre.

On écrit alors: $PC-; G+; L-; \exists \left(\begin{smallmatrix} PC+; (G = 1) \\ (G = 1); PC+ \end{smallmatrix} \right); G-; L+$.

Il peut également arriver qu'en répétant les expériences, on note sans conteste que pour certaines expériences PC passe son seuil avant d'observer le maximum des gènes de l'horloge, et pour d'autres seulement après, sans changer le comportement ultérieur. Dans ce cas, on sait que, quel que soit le chemin suivi, le résultat des expériences ne change pas.

On écrit alors: $PC-; G+; L-; \forall \left(\begin{smallmatrix} PC+; (G = 1) \\ (G = 1); PC+ \end{smallmatrix} \right); G-; L+$.

Enfin, on peut regrouper les observations de plusieurs expériences si l'on a constaté qu'une condition donnée modifie de manière systématique les chemins suivis. Par exemple, partant d'un état où $G = 0$ et $PC = 0$, on peut observer $G+$ puis $PC+$, et ensuite, selon qu'il y a ou non de la lumière, on peut ne plus observer de changement, ou bien observer une inhibition de G .

On écrit alors: $G+; PC+; (if (L = 1) then \varepsilon else G-)$ (où ε dénote le chemin vide).

En fait, tant que la lumière est présente, l'entrée de PC dans le noyau est inhibée, donc G n'est inhibé qu'à la disparition de la lumière. On peut alors par exemple écrire:

$G+; PC+; (while (L = 1) with (G = 1 \wedge PC = 1) do \exists \left(\begin{smallmatrix} \varepsilon \\ L- \end{smallmatrix} \right)); G-$

L'assertion $(G = 1 \wedge PC = 1)$ introduite par le mot-clef *with* est classiquement appelée un *invariant de boucle*. En logique de Hoare, cet invariant doit être fourni par

le programmeur, mais par bonheur, dans le cadre des réseaux de régulation biologique, il peut être calculé automatiquement et donc en pratique on ne s'en soucie pas, voir Bernot *et al.* (2018).

En fait, *while* est souvent utilisé pour décrire une portion de chemin où le comportement de certaines variables v_i est inconnu, jusqu'à aboutir à une

observation O donnée: $\dots ; (while (\neg O) with I do \exists \begin{pmatrix} \dots \\ v_i + \\ v_i - \\ \dots \end{pmatrix}) ; \dots$

Il est également utilisé pour prouver par l'absurde que certains états sont inatteignables: dans ce cas, O dénote l'ensemble d'états que l'on veut prouver inatteignables, l'ensemble des v_i est V tout entier, et l'algorithme de calcul automatique de l'invariant indique que I est contradictoire (aucun état ne vérifie l'invariant).

1.3.4. De la puissance des assertions

La possibilité d'utiliser les paramètres $K_{v,\omega}$ dans les assertions permet de *formaliser entièrement*, dans le langage des assertions, la théorie des réseaux de régulation biologique de René Thomas, comme montré étape par étape ci-dessous.

Puisque le langage des multiplexes est inclus dans le langage des assertions, exprimer qu'un multiplexe $m \in \Sigma^{-1}(v)$ est une ressource de la variable v s'écrit simplement φ_m (par définition, la formule du multiplexe est satisfaite dans un état donné si et seulement si m est ressource de v).

Dès lors, exprimer que $\omega \subseteq \Sigma^{-1}(v)$ est l'ensemble des ressources de v , c'est-à-dire que $K_{v,\omega}$ est le paramètre applicable, s'écrit:

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

(l'ensemble des ressources d'une variable dans un état donné est l'ensemble de ses multiplexes prédécesseurs dont la formule est satisfaite).

Puisqu'on veut formaliser l'approche de manière indépendante de l'état considéré, on ne sait pas par défaut quel ensemble ω est l'ensemble des ressources, cela dépend de l'état. Par conséquent formaliser la possibilité pour une variable v d'augmenter ou de diminuer doit considérer tous les cas possibles. C'est ce que traduisent respectivement les deux assertions suivantes:

$$\Phi_v^+ \equiv \bigwedge_{\omega \subseteq \Sigma^{-1}(v)} (\Phi_v^\omega \implies K_{v,\omega} > v) \quad \text{et} \quad \Phi_v^- \equiv \bigwedge_{\omega \subseteq \Sigma^{-1}(v)} (\Phi_v^\omega \implies K_{v,\omega} < v)$$

(si ω est l'ensemble de ressources de v alors le paramètre applicable $K_{v,\omega}$ doit « attirer v du côté voulu »).

Comme on le voit, la puissance du langage des assertions permet de formaliser la section 1.2 en quelques lignes! Pourtant, l'ensemble des symboles de V , M et $\mathcal{K}$ étant fini, et les entiers étant tous bornés par les bornes b_v , le langage des assertions peut être plongé dans une simple logique propositionnelle. C'est en fait la multiplicité des chemins au sein du graphe des transitions et le nombre exponentiel des paramètres qui apportent l'essentiel des difficultés d'identification des paramètres.

1.3.5. Une logique qui calcule la plus faible précondition

La logique de Hoare relève des mêmes principes que notre exemple de la section 1.3.1 mais l'objectif prioritaire est d'automatiser les raisonnements successifs pour les déléguer à l'ordinateur, donc les réduire à des manipulations de symboles. Comme il est techniquement plus simple de faire des substitutions que de la résolution, Dijkstra (1975) a montré qu'il est plus simple de *remonter* les chemins de la fin vers le début plutôt que de les parcourir en marche avant comme nous l'avons fait.

On part d'un *triplet de Hoare* de la forme $\{P\} \ c \ \{Q\}$ où P et Q sont les pré- et post-conditions mentionnées en section 1.3.2 et c est un chemin au sens de la section 1.3.3. Par exemple le triplet

$\{G = 0\} \ PC-; G+; L-; PC+; (G = 1); G-; L+ \ \{L = 1 \wedge PC = 1\}$ signifie que, d'une manière ou d'une autre, on s'est assuré que les gènes de l'horloge ne s'expriment pas notablement au début du protocole expérimental, puis qu'on a observé les évolutions de la figure 1.6, puis qu'on a noté que la lumière était éteinte et les complexes Per-Cry présents. Cela ne signifie pas que c'est le seul chemin possible mais seulement que celui-là a sans conteste déjà été observé.

Partant de la fin à la manière de Dijkstra, comment est-il possible que la lumière soit allumée ($L+$) et qu'ensuite on observe la post-condition ($L = 1 \wedge PC = 1$)?

– En premier lieu, il fallait que le paramètre applicable de L soit plus grand que la valeur courante de L pour effectuer $L+$, autrement dit que Φ_L^+ soit satisfait,

– mais aussi, puisque $L+$ signifie une transition qui transforme ($L \rightarrow L + 1$), il fallait qu'avant cette transition nous ayons $L + 1 = 1$, afin d'obtenir $L = 1$ après la transition, et bien sûr, puisque PC n'a pas évolué, il fallait aussi que $PC = 1$.

Inversement si ces deux conditions sont satisfaites, on peut franchir $L+$ et observer ensuite la post-condition $Q \equiv (L = 1 \wedge PC = 1)$. Ce raisonnement se généralise à toute variable v et à toute condition Q : pour que Q soit satisfaite après avoir franchi $v+$ il suffit que $Q' \equiv (\Phi_v^+ \wedge Q[v \leftarrow v + 1])$ le soit *juste avant* d'avoir franchi $v+$, où la notation $Q[v \leftarrow v + 1]$ dénote la formule Q dans laquelle on a remplacé toutes les occurrences de v par $v + 1$.

On a donc fabriqué une nouvelle post-condition $Q_1 \equiv (\Phi_L^+ \wedge L + 1 = 1 \wedge PC = 1)$ qui se simplifie en $(\Phi_L^+ \wedge L = 0 \wedge PC = 1)$ et il ne reste alors plus qu'à exploiter le

triplet

$\{G = 0\} \quad PC-; G+; L-; PC+; (G = 1); G - \{ \Phi_L^+ \wedge L = 0 \wedge PC = 1 \}$
 puisqu'on est certain qu'ensuite $L+$ sera franchissable et assurera la post-condition Q .
 On a ainsi écourté le chemin à traiter en supprimant $L+$.

Comme on s'en doute, on peut supprimer $G-$ du chemin selon le même principe en modifiant Q_1 en $Q_2 \equiv (\Phi_G^- \wedge Q_1 [G \leftarrow G - 1])$, ce qui donne (puisque G n'apparaît pas dans Q_1)

$$\{G = 0\} \quad PC-; G+; L-; PC+; (G = 1) \quad \{ \Phi_G^- \wedge \Phi_L^+ \wedge L = 0 \wedge PC = 1 \}$$

Pour supprimer une assertion (ici $(G = 1)$), c'est encore plus simple: puisqu'on veut qu'à la fois $(G = 1)$ et Q_2 soient satisfaites en ce point, il suffit de poser

$$Q_3 \equiv (G = 1) \wedge Q_2 \equiv G = 1 \wedge \Phi_G^- \wedge \Phi_L^+ \wedge L = 0 \wedge PC = 1$$

Comme on le voit, on réduit facilement la longueur du chemin à étudier, mais au prix d'une post-condition de plus en plus longue. En ce point, il serait bon de simplifier la formule. Des conditions $G = 1 \wedge L = 0 \wedge PC = 1$ on déduit que m_3 est satisfait et m_2 ne l'est pas, donc Φ_G^- est équivalent à $K_{G,\{ \}} < G$ et Φ_L^+ est équivalent à $K_{L,\{m_3\}} > L$, soit respectivement $K_{G,\{ \}} = 0$ et $K_{L,\{m_3\}} = 1$. Ainsi on aboutit à

$$\{G = 0\} \quad PC-; G+; L-; PC+ \quad \left\{ \begin{array}{l} K_{G,\{ \}} = 0 \wedge K_{L,\{m_3\}} = 1 \\ \wedge G = 1 \wedge L = 0 \wedge PC = 1 \end{array} \right\}$$

et continuer la démarche aboutirait naturellement aux mêmes contraintes sur les paramètres que la section 1.3.1.

D'un point de vue logique, nous avons appliqué successivement 4 *règles d'inférence*:

Incrémentatation: $\frac{\{ \Phi_v^+ \wedge Q[v \leftarrow v+1] \}}{v+ \{Q\}}$	Séquence: $\frac{\{Q''\} \quad c_1 \quad \{Q'\} \quad \{Q'\} \quad c_2 \quad \{Q\}}{\{Q''\} \quad c_1; c_2 \quad \{Q\}}$
Décrémentatation: $\frac{\{ \Phi_v^- \wedge Q[v \leftarrow v-1] \}}{v- \{Q\}}$	Assertion: $\frac{\{ \varphi \wedge Q \}}{\varphi \quad \{Q\}}$

Pour rappel, une règle d'inférence de la forme $\frac{H_1 \quad H_2 \quad \dots \quad H_n}{H}$ où les H_i et H sont des triplets de Hoare ou des assertions, se lit « pour prouver H , il suffit de prouver chacun des H_1 à H_n ».

Par exemple pour l'incrémentatation, n'ayant rien au-dessus de la barre d'inférence, il n'y a aucune preuve à faire, il suffit de construire $Q' \equiv \Phi_v^+ \wedge Q[v \leftarrow v+1]$.

En revanche, dans notre exemple, nous avons implicitement utilisé la règle de séquence qui requiert deux « sous-preuves »: pour prouver

$\{G = 0\} \quad PC-; G+; L-; PC+; (G = 1); G-; L+ \quad \{L = 1 \wedge PC = 1\}$
 nous avons posé $c_1 = (PC-; G+; L-; PC+; (G = 1); G)$ et $c_2 = L+$, de sorte qu'après avoir prouvé $\{ \Phi_L^+ \wedge L = 0 \wedge PC = 1 \} \quad L+ \quad \{L = 1 \wedge PC = 1\}$ il restait à prouver

$\{G = 0\} \quad PC-; G+; L-; PC+; (G = 1); G- \quad \{ \Phi_L^+ \wedge L = 0 \wedge PC = 1 \}$
 et ainsi de suite.

Cette stratégie qui consiste à choisir c_2 minimal à la fin du chemin et c_1 contenant tout le début du chemin est celle de Dijkstra et permet de calculer pas à pas les $Q_1, Q_2, \dots, Q_n$ qui, lorsque l'on a traité toutes les instructions (i.e. $c_1 = \varepsilon$), aboutissent à la *plus faible précondition* Q_n pour que le chemin soit faisable et permette d'observer la post-condition Q . Pour notre exemple, le lecteur pourra vérifier que l'on obtient:

$$Q_n \equiv Q_7 \equiv \left\{ \begin{array}{l} K_{PC, \{\}} = 0 \wedge K_{G, \{m_2\}} = 1 \wedge K_{L, \{\}} = 0 \wedge K_{PC, \{m_1\}} = 1 \\ \wedge K_{G, \{\}} = 0 \wedge K_{L, \{m_3\}} = 1 \wedge G = 0 \wedge L = 1 \wedge PC = 1 \end{array} \right\}$$

Dans le cadre des programmes impératifs en informatique, une fois la plus faible précondition Q_n obtenue, pour qu'un triplet de Hoare $\{P\} c \{Q\}$ soit prouvé, il ne reste plus qu'à prouver que $P \Rightarrow Q_n$ où Q_n est le résultat de la dernière étape (lorsque $c_1 = \varepsilon$).

Ceci s'exprime de nouveau par une règle d'inférence: *Programme vide*: $\frac{P \Rightarrow Q}{\{P\} \varepsilon \{Q\}}$.

Dans le cadre des chemins expérimentaux en biologie, on *sait* que P est une connaissance initiale, donc que P est satisfaite. Autrement dit P est en fait une assertion comme une autre, et l'on applique alors la règle d'inférence plus forte:

Chemin vide: $\frac{P \wedge Q}{\{P\} \varepsilon \{Q\}}$.

Pour notre exemple de triplet de Hoare, il sera donc prouvé si et seulement si

$$G = 0 \wedge \left\{ \begin{array}{l} K_{PC, \{\}} = 0 \wedge K_{G, \{m_2\}} = 1 \wedge K_{L, \{\}} = 0 \wedge K_{PC, \{m_1\}} = 1 \\ \wedge K_{G, \{\}} = 0 \wedge K_{L, \{m_3\}} = 1 \wedge G = 0 \wedge L = 1 \wedge PC = 1 \end{array} \right\}$$

c'est-à-dire si et seulement si:

$$\left\{ \begin{array}{l} K_{PC, \{\}} = 0 \wedge K_{G, \{m_2\}} = 1 \wedge K_{L, \{\}} = 0 \wedge K_{PC, \{m_1\}} = 1 \\ \wedge K_{G, \{\}} = 0 \wedge K_{L, \{m_3\}} = 1 \wedge G = 0 \wedge L = 1 \wedge PC = 1 \end{array} \right\}$$

ce qui impose la valeur des paramètres et limite aux états initiaux où la lumière est allumée, les gènes de l'horloge faiblement exprimés et les complexes Per-Cry présents. On y reviendra mais commençons par compléter notre stratégie générale en marche arrière pour traiter toutes les primitives du langage des chemins.

Lorsque cette stratégie rencontre un quantificateur, c'est-à-dire que c_2 est de la forme $\exists \left(\begin{smallmatrix} c' \\ c'' \end{smallmatrix} \right)$ ou bien $\forall \left(\begin{smallmatrix} c' \\ c'' \end{smallmatrix} \right)$, la première forme signifie qu'on ne sait pas par quel chemin c' ou c'' on est passé, donc que l'une *ou* l'autre des préconditions respectivement calculées Q' ou Q'' doit s'appliquer, alors que la seconde forme affirme que le résultat est obtenu quel que soit le chemin choisi, donc que la *conjonction* des deux préconditions doit s'appliquer. Ceci se formalise simplement:

$$\textbf{Existentiel: } \frac{\{Q'\} c' \{Q\} \quad \{Q''\} c'' \{Q\}}{\{Q' \vee Q''\} \exists \left(\begin{smallmatrix} c' \\ c'' \end{smallmatrix} \right) \{Q\}} \quad \textbf{Universel: } \frac{\{Q'\} c' \{Q\} \quad \{Q''\} c'' \{Q\}}{\{Q' \wedge Q''\} \forall \left(\begin{smallmatrix} c' \\ c'' \end{smallmatrix} \right) \{Q\}}$$

Les règles d'inférence pour franchir une affectation de variable, un *while* ou un *if* sont quant-à elles directement celles de Tony Hoare:

Affectation: $\frac{}{\{Q[v \leftarrow n]\} v := n \{Q\}}$

Boucle: $\frac{\{a \wedge I\} c \{I\} \quad \neg a \wedge I \Rightarrow Q}{\{I\} \text{ while } a \text{ with } I \text{ do } c \{Q\}}$

Conditionnelle:
$$\frac{\{Q'\} \text{ c' } \{Q\} \quad \{Q''\} \text{ c'' } \{Q\}}{\{(a \wedge Q') \vee (\neg a \wedge Q'')\} \text{ if } a \text{ then c' else c'' } \{Q\}}$$

L'invariant I d'une boucle *while* est une assertion qui, si elle est satisfaite à l'entrée d'un « tour de boucle », restera satisfaite à l'issue du tour de boucle; on est ainsi assuré que I reste vraie jusqu'à la sortie du *while*. Encore faut-il qu'initialement I soit satisfaite à l'entrée de la boucle, c'est pourquoi c'est la précondition du triplet de Hoare dans la règle de boucle. La post-condition doit à son tour être une conséquence de ce qui est connu à la sortie du *while*, à savoir que I est satisfaite et que l'on sort du *while*, donc que l'assertion a est fausse: $\neg a \wedge I \implies Q$. Enfin $\{a \wedge I\} \text{ c } \{I\}$ se contente de vérifier que I est bien un invariant: si l'on fait un tour de boucle (i.e. si a) alors I est préservé au travers du corps de la boucle (c). En fait, comme déjà mentionné, I peut être calculé automatiquement pour les réseaux de régulation biologique et donc cette dernière sous-preuve est automatiquement assurée par l'algorithme de calcul de I .

Lorsqu'on rencontre un *if*, de deux choses l'une: si a est vraie alors on passe par c' , et Q' doit donc être également satisfaite ($a \wedge Q'$), ou bien si a est fausse alors on passe par c'' et Q'' doit être satisfaite ($\neg a \wedge Q''$). C'est exactement ce que formalise la règle de conditionnelle.

Remarquons enfin que pour raisonner et simplifier les assertions, on utilise des connaissances sur les bornes et sur l'addition ou la soustraction. Par exemple la réduction de $L + 1 = 1$ en $L = 0$ a utilisé le fait que 0 est neutre et qu'on peut simplifier à droite, et la simplification de $K_{L,\{m_3\}} > 0$ en $K_{L,\{m_3\}} = 1$ a utilisé la connaissance de la borne $b_L = 1$. Ce sont là autant d'axiomes ajoutés aux règles précédentes; ils sont élémentaires et en nombre fini.

Toutes ces règles et axiomes constituent la *logique de Hoare génétiquement modifiée* et le résultat majeur de Bernot *et al.* (2018) est que cette logique est correcte, complète et décidable. Autrement dit:

- la stratégie de calcul de la plus faible précondition pour un triplet de Hoare $\{P\} \text{ c } \{Q\}$, décrite dans cette section, est un algorithme qui établit automatiquement des contraintes C sur les interprétations σ des paramètres $K_{v,\omega}$ et sur les états initiaux η ,

- on peut donc considérer l'ensemble des interprétations $\sigma \in \mathcal{S}_\Sigma$ (définition 1.8) et des états $\eta \in \zeta$ (définition 1.2) tels que $\eta(\sigma(C))$ est satisfaite: cela détermine un ensemble d'identifications possibles des paramètres $\mathcal{S}_C \subseteq \mathcal{S}_\Sigma$, pour lesquels l'ensemble des états initiaux possibles $\zeta_{C,\sigma} \subseteq \zeta$ est non vide,

- cet ensemble d'identifications possibles $\mathcal{S}_C$ détermine *exactement* l'ensemble des réseaux de régulation biologique tels que, dans le graphe de transitions, tous les états initiaux de $\zeta_{C,\sigma}$ satisfont P , peuvent suivre le chemin c , et à l'issue de ce chemin, Q est satisfaite.

Rappelons toutefois que la logique de Hoare génétiquement modifiée n'est applicable que si toutes les variables du graphe d'influences sont observées avec une

résolution temporelle suffisante, ce qui impose des conditions expérimentales assez drastiques. Lorsque les observations sont partielles, on devra utiliser une autre approche formelle: la logique temporelle, qui fait l'objet de la section suivante.

Enfin, même si sur des exemples de taille réelle les chemins connus expérimentalement ne suffisent généralement pas à déterminer la valeur de tous les paramètres, la connaissance de quelques chemins suffit à réduire considérablement l'espace de recherche des paramétrages possibles, comme on le verra en 1.5. Cela permet ensuite d'exploiter plus efficacement les connaissances temporelles (section suivante) en limitant le nombre de modèles à tester.

1.4. Logique temporelle et CTL

Bon nombre de connaissances biologiques ne se traduisent pas directement sous forme de triplets de Hoare: il est souvent difficile de mesurer toutes les variables avec une résolution temporelle suffisante et bon nombre de connaissances biologiques sont d'une autre nature, telles par exemple la connaissance de phénomènes épigénétiques, d'homéostasie, d'atteignabilité ou de non atteignabilité de certains comportements, d'événements qui arrivent toujours après d'autres (mais pas nécessairement juste après), *etc.*

Pour formaliser ces connaissances générales, les logiques classiques comme la logique propositionnelle ou la logique du premier ordre ne sont pas bien adaptées car, dans ces logiques, une propriété donnée est soit toujours vraie soit toujours fausse. Pour formaliser des états changeant au cours du temps, il est préférable de disposer de modalités temporelles en plus des connecteurs logiques, donc de faire appel à une logique temporelle. Nous avons choisi dans Bernot *et al.* (2004) la logique CTL (*Computation Tree Logic*, Clarke and Emerson (1981)) en raison de l'efficacité de son algorithme de model checking (McMillan (1993), Huth and Ryan (2000)), et de sa capacité à capturer les choix non déterministes de la théorie de René Thomas. Comme on va le voir, les formules CTL sont extrêmement utiles pour une identification automatique des paramètres.

1.4.1. CTL et model checking

Si l'on considère une connaissance biologique de plus haut niveau que l'observation pas à pas de toutes les variables, telle que par exemple « *l'expression des gènes de l'horloge oscille* » on peut sans grande difficulté l'exprimer en termes d'états: « *Si $G = 0$ alors plus tard $G = 1$ et si $G = 1$ alors plus tard $G = 0$* ». Cette formulation n'est pas tout à fait complète: comme on l'a vu, plusieurs paramètres peuvent être applicables dans un état donné et le futur d'un état est donc non déterministe, de sorte qu'on doit encore préciser si ces deux affirmations concernent tous les chemins possibles ou seulement l'existence d'un chemin.

Plus généralement, partant d'un état initial donné, l'ensemble des états suivants n'est pas nécessairement réduit à un singleton car plusieurs transitions peuvent éventuellement partir de cet état (une transition par paramètre applicable dans cet état). Le futur d'un état initial donné peut donc être représenté sous la forme d'un arbre: la logique temporelle CTL, qui signifie *Computation Tree Logic*, est bien adaptée pour exprimer des propriétés sur de tels graphes de transitions non déterministes.

En l'absence de prémisses sur le comportement de la lumière, il peut être sage de se limiter à l'existence de chemins et dans ce cas on écrira en CTL:

$$(G = 0 \Rightarrow EF(G = 1)) \wedge (G = 1 \Rightarrow EF(G = 0))$$

Dans cette formule, $EF(\dots)$ signifie « il existe un chemin tel que dans le futur G deviendra égal à 1 (resp. 0) ». La prémisses $G = 0$ (resp. $G = 1$) a pour conséquence de restreindre la vérification de $EF(\dots)$ aux arbres dont l'état initial (la racine) vérifie $G = 0$ (resp. $G = 1$), soit 4 états initiaux pour chacune des implications puisque L et PC sont booléens. Les atomes $G = 0$ et $G = 1$ ne sont que de simples propositions; il en existe un nombre fini en raison de la borne b_v pour toute variable $v \in V$. S'il n'y avait pas de modalités temporelles nous serions donc dans une logique propositionnelle.

Les modalités temporelles, quant-à elles, sont toujours formées par deux lettres:

1) La première lettre est un quantificateur. « E » signifie *il existe un chemin dans l'arbre tel que...* et « A » signifie *pour tous les chemins de l'arbre*.

2) La seconde lettre concerne l'écoulement du temps le long de chaque chemin considéré:

- « F » signifie *dans le futur* il y aura un état du chemin qui satisfait la propriété sous sa portée, mais sans préciser quand exactement,

- « X » signifie *l'état suivant* c'est-à-dire que l'état juste après l'état courant dans le chemin satisfera la propriété sous sa portée,

- « G » signifie *globalement* c'est-à-dire que la propriété est satisfaite dans tous les états le long du chemin,

- enfin « U » signifie *jusqu'à*, elle requiert deux propriétés φ_1 et φ_2 sous sa portée, et elle signifie que φ_1 est et restera satisfaite tout le long du chemin jusqu'à ce que φ_2 le devienne à un moment donné. À noter que φ_2 doit devenir satisfaite dans le chemin, c'est-à-dire que cela implique « F » pour φ_2 .

Les deux lettres sont indissociables en CTL, de sorte que, par exemple, F seul n'est pas une modalité; on doit préciser soit EF , soit AF . La syntaxe du *until* place la lettre U en infixe entre φ_1 et φ_2 mais le quantificateur reste requis: on doit écrire soit $E[\varphi_1 U \varphi_2]$, soit $A[\varphi_1 U \varphi_2]$.

Considérons la formule: $(PC = 1 \wedge L = 1) \Rightarrow AF(AG(PC = 1))$. Elle signifie que partant d'un état quelconque tel que $PC = 1$ avec la lumière allumée, le

complexe PC finira par toujours rester présent, quel que soit le chemin choisi. Une condition suffisante évidente pour que cette formule soit satisfaite est que $K_{PC,\{\}} = K_{PC,\{m_1\}} = 1$. Mais la question qui intéresse les modélisateurs et les biologistes est *est-ce le seul cas?*... Le lecteur averti pourra vérifier que non (avec $K_{PC,\{\}} = 0$ mais des paramètres $K_{G...}$ qui attirent G vers 1), cependant pour des graphes d'influences de plus grande taille, il est impossible de répondre sans l'aide d'un environnement logiciel qui parcourt de manière systématique toutes les interprétations correctes des paramètres.

De même la formule $AF(AG(L = 1)) \Rightarrow AF(AG(G = 1))$ signifie que si l'on allume la lumière à un moment donné sans jamais l'éteindre, alors les gènes de l'horloge finiront par s'exprimer de manière constante. On y reviendra dans la sous-section suivante.

L'avantage de CTL, outre ses quantificateurs qui permettent de prendre en compte le non déterminisme des chemins de la théorie de Thomas, est qu'elle autorise des algorithmes de *model checking* très performants. Ils prennent en entrée un automate et une formule CTL, et retournent un contre-exemple s'il existe un chemin dans l'automate qui contredit la formule. Les performances des model checkers sont remarquables parce qu'ils ne sont pas principalement fondés sur des parcours de chemins mais un parcours de la formule à vérifier. À ce propos, un point technique qui a historiquement aidé à ces performances est que *le futur contient le présent* en CTL. Ainsi, si l'état initial satisfait une propriété φ , alors $AF(\varphi)$ et $EF(\varphi)$ sont automatiquement satisfaits même si φ n'est plus jamais satisfaite dans le futur strict. Si l'on souhaite ne faire appel qu'au futur strict, il suffit simplement d'ajouter AX ou EX en début de formule. De la même façon, si l'état courant satisfait φ_2 alors $E[\varphi_1 U \varphi_2]$ et $A[\varphi_1 U \varphi_2]$ sont satisfaites même si φ_1 est non satisfaite dans l'état courant.

Les graphes de transitions construits dans la définition 1.7 à partir d'une interprétation des paramètres $K...$ sont des automates, donc le model checking permet sans difficulté de retourner un contre-exemple si cette interprétation autorise un arbre qui ne satisfait pas la formule CTL soumise. S'il n'existe aucun contre-exemple, cela signifie que l'interprétation des paramètres $K...$ qui a permis de construire le graphe de transitions est en accord avec les connaissances biologiques exprimées en CTL.

Néanmoins il faut être conscient que ce qui permet d'étudier un système biologique *n'est pas* la connaissance d'une interprétation des paramètres qui satisfasse la conjonction de toutes les formules (et les triplets de Hoare) issues de la connaissance biologique. En effet, une connaissance supplémentaire issue d'expériences nouvelles peut à tout moment contredire l'interprétation choisie et il faudrait recommencer l'identification des paramètres. Ce qui permet d'accompagner les biologistes dans leur recherche, c'est la connaissance *exhaustive de toutes les interprétations* qui satisfont les connaissances biologiques. C'est l'objet de la section

suivante (TotemBioNet) qui automatise cette recherche exhaustive. De cette façon, une connaissance nouvelle ajoute simplement une formule à la conjonction de formules courantes et réduit donc l'ensemble des interprétations compatibles avec les connaissances. Si cet ensemble devient vide, il faut alors remettre en cause le graphe d'influences.

1.4.2. Fair path CTL

Revenons à la formule $AF(AG(L = 1)) \Rightarrow AF(AG(G = 1))$. Une condition suffisante pour atteindre ce comportement est que $K_{G, m_2} = 1$. On pourrait penser que c'est évident car si $L = 1$ de manière constante à partir d'un certain moment, alors m_2 deviendra une ressource constante de G à ce même moment, donc G sera « attiré » de manière constante par 1, deviendra donc égal à 1 à un moment ou un autre et y restera.

En fait, ce n'est absolument pas évident en CTL et le raisonnement précédent est mis en défaut dans de nombreux cas lorsqu'un modèle contient beaucoup de variables! Pour s'en rendre compte, imaginons simplement le modèle de la figure 1.7 dans

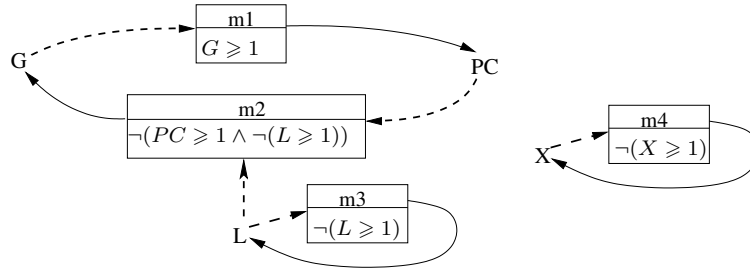


Figure 1.7. Lorsque $K_{X, \{m4\}} = 1$ et $K_{X, \{\}} = 0$ l'oscillation de X est toujours possible: lorsque X vaut 0 il est attiré vers $K_{X, \{m4\}} = 1$ et lorsqu'il vaut 1, il est attiré vers $K_{X, \{\}} = 0$. Il existe donc un chemin (non équitable) où X est la seule variable du réseau qui évolue et L , G et PC restent constantes le long de ce chemin.

lequel on a ajouté une variable X munie d'une auto-inhibition active, c'est-à-dire $K_{X, \{m4\}} = 1$ et $K_{X, \{\}} = 0$. Quel que soit l'état initial, donc en particulier lorsque $G = 0$, il existe toujours un chemin où X oscille entre 0 et 1, et où les autres variables ne se mettent jamais à jour. Ce chemin est *non équitable* en ce sens qu'il passe une infinité de fois par un état où, entre autres, la transition ($G = 0 \rightarrow G = 1$) est franchissable, mais n'est jamais empruntée.

Ainsi, dans ce modèle avec une variable supplémentaire, le model checking trouvera un contre-exemple à la formule $AF(AG(L = 1)) \Rightarrow AF(AG(G = 1))$,

mais ce chemin est en fait, de toute évidence, un artefact dû à la sémantique de CTL, biologiquement non crédible. Il ne faut pas croire que cet artefact résulte de la non connectivité du graphe d'influences: sans X , il existe néanmoins des chemins où L oscille entre 0 et 1 sans que G et PC ne se mettent jamais à jour!

Il en résulte que, dans le cadre des réseaux de régulation biologique, on doit généralement utiliser une version de CTL restreinte aux chemins équitables, c'est-à-dire où les quantificateurs A et E ignorent les chemins non équitables. Un chemin sera dit *non équitable* s'il passe une infinité de fois dans un état d'où part une transition qu'il n'emprunte jamais. On définit donc CTL' (*fair path* CTL) en remplaçant simplement les quantificateurs A et E par deux autres quantificateurs A' et E' qui signifient respectivement *pour tous les chemins équitables* et *il existe un chemin équitable*, avec les mêmes règles syntaxiques que pour A et E .

En fait, il existe plusieurs versions de « fair CTL » dans la littérature mais, à notre connaissance, aucune n'adopte cette définition. C'est éventuellement dû au fait que notre fair path CTL n'augmente pas le pouvoir d'expression de CTL: Adrien Richard (2008) démontre assez simplement qu'il existe une traduction récursive des formules de CTL' en formules de CTL (qui préserve la satisfaction des formules, naturellement). Ainsi on continue à bénéficier de la puissance des algorithmes de model checking. Cette traduction automatique est incluse dans la plate-forme *TotemBioNet* où l'on peut choisir d'écrire une formule en CTL d'origine ou bien en fair path CTL.

1.5. TotemBioNet

TotemBioNet est une plate-forme dédiée à l'identification des paramètres pour la formalisation discrète de René Thomas. *TotemBioNet* exploite la complémentarité des approches basées sur la logique de Hoare et la logique temporelle. Le bénéfice est double. D'une part, le pouvoir d'expression est renforcé. Pour un même cas d'étude, il est permis d'exprimer des propriétés d'atteignabilité telles que la multi-stationnarité ou l'homéostasie par la logique temporelle, ainsi que des propriétés de chemins par la logique de Hoare. D'autre part, les performances sont améliorées car la plus faible pré-condition calculée par logique de Hoare sert à réduire l'espace de recherche des paramétrages. Nous commençons dans cette section par présenter l'outillage de *TotemBioNet*: principalement *Hoare-fol* qui construit la plus faible pré-condition de Hoare, *SMBioNet* qui filtre les paramétrisations qui mènent à un graphe de transitions satisfaisant la formule CTL et enfin l'outil qui implémente la méthode d'intégration de ces deux approches. Nous montrons ensuite sur deux exemples classiques les performances de la plate-forme.

1.5.1. Outillage

1.5.1.1. Hoare-fol

Hoare-fol écrit par Maxime Folshette, voir Folschette (2011), met en œuvre le calcul de plus faible pré-condition présenté dans la section 1.3. Les mécanismes sont les suivants ^{5 6}:

- les règles d'inférence de la logique de Hoare génétiquement modifiée sont appliquées pour calculer la plus faible pré-condition wp ,
- si une pré-condition est fournie, elle est propagée pour simplifier wp en swp grâce à des règles de simplifications formelles comme par exemple $a \wedge False \Rightarrow False$.

En toute généralité, le formalisme présenté en 1.3 ne requiert pas la connaissance complète des pré- et post-conditions, qui dépendent de ce qui a été observé par le biologiste. Les pré- et post-conditions peuvent être présentes et complètes (partant d'un état totalement spécifié, on a observé une trace et la valeur finale de chacune des variables a été mesurée), absentes (on a simplement observé une trace), ou peuvent porter sur un sous-ensemble strict des variables de la trace (l'état initial et/ou l'état final n'a été que partiellement observé, autrement dit, n'est spécifié que pour un sous-ensemble des variables de la trace).

D'un point de vue pratique, nous nous limitons dans *TotemBioNet* à des traces de Hoare dont la pré-condition fournit une information complète sur les variables de la trace. Avec cette restriction, la formule swp obtenue après simplification est:

- 1) réduite à $False$ par simplification, si le triplet est inconsistant, ou bien
- 2) une conjonction de formules atomiques (*i.e.* de la forme $K_v \leq s$ ou $\neg(K_v \leq s)$) qui caractérise les paramétrages dont la dynamique est cohérente avec la trace observée, ou encore
- 3) une disjonction de formules de la forme 2, si la trace contient des quantificateurs existentiels ou des conditionnelles.

1.5.1.2. SMBioNet

SMBioNet est un outil développé par Adrien Richard qui a été utilisé pour modéliser et identifier les paramètres pour de nombreux cas d'études comme par exemple la production de mucus chez *Pseudomonas aeruginosa* étudiée par Pérès and Comet (2003), le contrôle de l'immunité du bactériophage lambda analysé par Richard *et al.* (2006), l'analyse des phases d'apoptose et de croissance de la queue du têtard proposée par Khalis, Comet, Richard and Bernot (2009), et plus récemment

5. Le traitement des boucles n'est pas abordé ici car il est partiel dans la version actuelle.

6. *Hoare-fol* permet également de générer un programme ASP qui calcule les paramétrages satisfaisant le triplet de Hoare, mais cette fonctionnalité n'a pas été intégrée car la version de *Clingo* utilisée est obsolète, et des optimisations sont nécessaires.

pour appuyer l'expertise du biologiste pour l'identification des paramètres d'un modèle de la régulation du métabolisme énergétique, Cf. Khoodeeram *et al.* (2017). *SMBioNet* est également un élément clef pour chercher le signe des célérités dans les modélisations hybrides (section 1.6). *SMBioNet* permet d'énumérer de façon efficace les paramétrages possibles d'un réseau de régulation et de filtrer les paramétrages qui valident une formule temporelle (Cf. 1.4). Plus précisément, *SMBioNet* prend en entrée:

- la description d'un graphe d'influences avec multiplexes,
- des informations éventuelles sur les paramètres comme par exemple pour fixer le comportement de Zeitgeber de la lumière (voir 1.2.2.5),
- une propriété en logique CTL.

SMBioNet renvoie en sortie l'ensemble des interprétations des paramètres qui mènent à une dynamique qui valide la formule CTL passée en argument. Il est à noter que *SMBioNet* ne cherche pas *un* paramétrage mais *tous* les paramétrages qui vérifient la propriété. Ceci est en effet cohérent avec la démarche de dialogue entre le biologiste et le bio-informaticien. L'objectif est de modéliser au mieux l'hypothèse du biologiste afin de le conforter dans le fait qu'il est effectivement intéressant de mener une expérience *in-vivo* ou *in-vitro*. Si l'on obtient un ensemble de paramétrages vide, l'hypothèse est fausse (pour le graphe d'influences considéré), si l'ensemble des paramétrages est trop grand, l'hypothèse a éventuellement peu d'intérêt (ou les connaissances sont incomplètement formalisées), si l'ensemble des paramétrages est petit, cela peut indiquer que cette hypothèse est intéressante à vérifier de façon expérimentale.

SMBioNet procède par énumération de tous les paramétrages possibles et génère à la volée l'automate qui correspond au graphe de transitions asynchrone du paramétrage courant. Cet automate et la formule CTL sont traduits et résolus par le solveur *NuSMV* (voir Cimatti *et al.* (2002)). L'énumération des paramètres est optimisée de la façon suivante:

- elle tient compte des informations sur les paramètres, valeurs ou bornes, fournies par le modélisateur dans le bloc de déclaration PARA, afin de réduire les domaines et par conséquent l'espace de recherche,
- elle respecte les conditions de monotonicité de Snoussi (1989)⁷:

$$\forall v \in V, \quad \forall \omega \subseteq \Sigma^{-1}(v), \quad \forall \omega' \subseteq \Sigma^{-1}(v), \quad \omega \subseteq \omega' \Rightarrow K_{v,\omega} \leq K_{v,\omega'}$$
- pour les variables v auto-réglées c'est à dire telles que v participe à un multiplexe contrôlant lui-même v , toutes les valeurs des $K_{v,\omega}$ ne sont pas énumérées (Richard (2006)).

7. Cette condition est nécessaire pour que le modèle discret soit en accord avec un modèle basé sur un système différentiel linéaire par morceau.

Notons que les performances de *SMBioNet* sont très fortement liées au nombre de modèles à énumérer, la vérification de la formule CTL pour un graphe de transitions fixé est extrêmement rapide car la formule à vérifier est en général petite.

1.5.1.3. *TotemBioNet*

TotemBioNet intègre *hoare-fol* et une version étendue de *SMBioNet*. *TotemBioNet* permet de déclarer des *variables d'environnement* qui figent l'état dans lequel on souhaite analyser la dynamique du système. Par exemple, pour le cycle circadien, L peut être vue comme une variable d'environnement qui sera figée à 0 sur tous les états pour étudier les expériences en obscurité totale. Ceci permet de simplifier les formules CTL en les allégeant du contexte d'observation. *TotemBioNet* fournit aussi une macro de traduction des formules CTL' en CTL (voir section 1.4.2). Ainsi le modélisateur exprime la propriété temporelle telle qu'il la conçoit, en sachant que l'outil l'interprète avec une sémantique adaptée aux comportements biologiques. Un bloc de déclaration *Hoare* permet d'ajouter une trace de Hoare dont la précondition doit être *complète* c'est à dire fixer les valeurs de toutes les variables de la trace. Dans ce cas, l'outil *hoare-fol* est invoqué et calcule la plus faible pré-condition simplifiée *swp* qui est utilisée pour filtrer les paramétrages de la façon suivante:

- si *swp* est *False* aucun paramétrage ne satisfait la trace,
- si *swp* est une conjonction d'atomes de la forme $(K_v \leq s)$ où $\neg(K_v \leq s)$, alors le domaine des paramètres présents dans *swp* est réduit,
- si *swp* contient des disjonctions, seuls les paramétrages qui satisfont *swp* sont considérés pour construire l'automate vérifié par *NuSMV*.

Les trois points ci-dessus permettent de réduire considérablement l'espace de recherche des paramétrages comme illustré dans les deux exemples des sections suivantes.

1.5.2. Exemple 1: croissance et apoptose de la queue du têtard

Ce premier exemple illustre l'intérêt de *TotemBioNet* dans le cas où l'ordre des atomes de la trace est bien connu. Le cas d'étude concerne la transformation du têtard en grenouille et plus particulièrement les phases de croissance et d'apoptose de la queue du têtard. Nous nous concentrons ici sur les propriétés à vérifier sur la dynamique du système et l'intérêt de la combinaison des deux approches. Les détails de la modélisation peuvent être trouvés dans Khalis, Comet, Richard and Bernot (2009) et Khalis (2010).

Le graphe d'influences étudié est celui de la figure 1.8. Le réseau est régulé par deux types d'hormones thyroïdiennes la tri-iodothyronine (T3) et la thyroxine (T4) elles-mêmes régulées par deux types de déiodinases notées D2 et D3. Ces deux enzymes ôtent l'iode présente dans les hormones thyroïdiennes ce qui les rendent

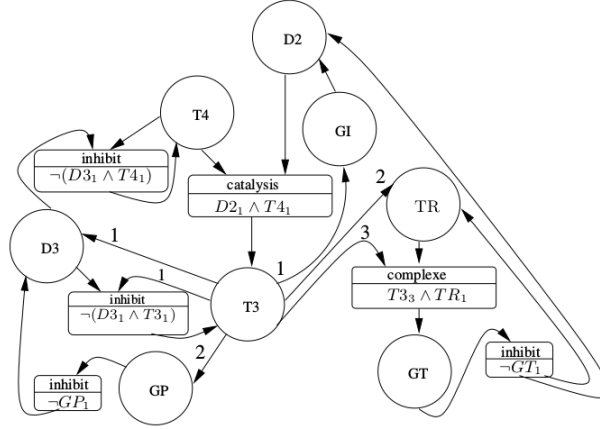


Figure 1.8. *Graphe d'influences de la résorption de la queue du têtard.*
 Dans cette figure et la suivante, un arc (u, v) étiqueté par n (resp. $-n$) correspond au multiplexe de formule $(u \geq n)$ (resp. $\neg(u \geq n)$) et dans les formules de multiplexe, u_n est un raccourci pour $(u \geq n)$.

actives ou non dans le mécanisme d'apoptose. Les régulateurs de D2 et D3 forment les groupes des gènes précoces GP, des gènes intermédiaires GI, et des gènes tardifs GT, appelés ainsi parce qu'ils s'expriment dans cet ordre. Ils permettent la mise en place effective de l'apoptose. TR est un gène précoce, isoforme du récepteur nucléaire de T3 qui joue un rôle prépondérant et apparaît donc de façon isolée dans la modélisation. Pour les seuils, la borne maximale de T3 est 4, celle de T2 est 3 et les autres variables sont booléennes (0 ou 1).

1.5.2.1. Comportement temporel

Khalis, Comet, Richard and Bernot (2009) décrivent deux phases distinctes:

- 1) la phase de croissance, qui part de l'état $(T3 = 0, T4 = 0, D2 = 0, D3 = 1, GI = 0, GP = 0, TR = 0, GT = 0)$ jusqu'au *climax* $(T3 = 3, T4 = 1, D2 = 1, D3 = 0, GI = 1, GP = 1, TR = 1, GT = 0)$,
- 2) la phase d'apoptose, qui part du climax jusqu'à ce que GT passe à 1.

Le comportement temporel est tel que, en partant de $(T3 = 0, T4 = 0, D2 = 0, D3 = 1, GI = 0, GP = 0, TR = 0, GT = 0)$, on atteint toujours le climax, et à partir du

climax, on peut atteindre l'apoptose. Cela se traduit par la formule CTL suivante:

$$\left(\begin{pmatrix} T3 = 0 \wedge \\ T4 = 0 \wedge \\ D2 = 0 \wedge \\ D3 = 1 \wedge \\ GI = 0 \wedge \\ GP = 0 \wedge \\ TR = 0 \wedge \\ GT = 0 \end{pmatrix} \Rightarrow AF \begin{pmatrix} T3 = 3 \wedge \\ T4 = 1 \wedge \\ D2 = 1 \wedge \\ D3 = 0 \wedge \\ GI = 1 \wedge \\ GP = 1 \wedge \\ TR = 1 \wedge \\ GT = 0 \end{pmatrix} \right) \wedge \left(\begin{pmatrix} T3 = 3 \wedge \\ T4 = 1 \wedge \\ D2 = 1 \wedge \\ D3 = 0 \wedge \\ GI = 1 \wedge \\ GP = 1 \wedge \\ TR = 1 \wedge \\ GT = 0 \end{pmatrix} \Rightarrow AF(GT = 1) \right)$$

1.5.2.2. Trace observée

En s'appuyant sur des profils d'expression de la phase d'apoptose, le triplet de Hoare suivant a été établi:

$$\left\{ \begin{array}{l} T4 = 1, d3 = 1, gi = 0, gp = 0, \\ tr = 0, gt = 0, d2 = 0, T3 = 1 \end{array} \right\} gi+; d2+; T3+; tr+; T3+; gp+; d3-; gt+ \{ \}$$

Notons qu'il n'y a pas de post-condition: partant d'un état initial, on a simplement observé une trace.

1.5.2.3. Résultats de TotemBioNet

Dans Khalis, Comet, Richard and Bernot (2009), des connaissances sur les paramètres ont été exploitées et ont permis d'identifier avec *SMBioNet* 18 paramétrages cohérents avec la littérature.

Sans aucune information sur les paramètres, il y a au total 1 749 600 paramétrages respectant les conditions de Snoussi. *SMBioNet* résout en moyenne 150 paramétrages par seconde, il faudrait de l'ordre de 3h30 pour trouver tous les paramétrages⁸.

La plus faible pre-condition simplifiée est calculée instantanément et correspond à la formule: $(K_{gi+T3_1} > 0) \wedge (K_{d2+gi_1+\neg gt_1} > 0) \wedge (K_{T3+catalys} > 1) \wedge (K_{tr+T3_2+\neg gt_1} > 0) \wedge (K_{T3+catalys} > 2) \wedge (K_{gp+T3_2} > 0) \wedge (K_{d3+T3_1} < 1) \wedge (K_{gt+complex} > 0)$.

Il s'agit d'une conjonction d'atomes puisque la trace ne contient pas d'instruction conditionnelle ni de quantificateur existentiel. Chaque inégalité sur un paramètre K est utilisée par *TotemBioNet* pour réduire son domaine. Il reste ainsi 180 000 paramétrages à explorer, le nombre de paramétrages a donc été divisé par 10 environ. Parmi ces 180 000 paramétrages, *TotemBioNet* sélectionne 40620 paramétrages qui satisfont la formule CTL en 23m22s.

8. Les temps sont donnés pour illustrer l'intérêt de la combinaison des méthodes. Ils ont été mesurés sur un Intel® Core i7-7600U, CPU 2.80GHz × 4, 16 Gio de mémoire.

1.5.3. Exemple 2: cycle cellulaire simplifié des mammifères

Ce deuxième exemple illustre le pouvoir d'expression de la logique de Hoare quand les informations sur la trace sont partielles. Le cas d'étude est le cycle cellulaire des mammifères dont un modèle simplifié à cinq variables a été présenté par Behaegel *et al.* (2015). Ce cycle contient des points de synchronisation bien identifiés qui permettent de le décomposer en phases successives. Si la succession des phases est connue, l'ordre précis des transitions à l'intérieur d'une phase ne l'est pas. Pour chaque phase, il y a donc autant de possibilités que de permutations des transitions de la phase. Le quantificateur existentiel de la logique de Hoare permet d'exprimer qu'une au moins de ces permutations constitue un chemin possible.

Le graphe d'influences proposé par Behaegel *et al.* (2015) est présenté en figure 1.9. Ce graphe représente les aspects moléculaires du cycle cellulaire en tenant compte des points de contrôle et des mécanismes de surveillance de la mitose spécifiques aux mammifères. Les variables représentent des protéines ou des complexes. Par exemple, *sk* représente les complexes *Cyclin E/Cdk2* et *Cyclin H/Cdk7*. Nous nous concentrons ici sur l'intérêt de la combinaison des approches par *model checking CTL* et par *logique de Hoare* et renvoyons le lecteur à Behaegel *et al.* (2015) pour les détails de modélisation.

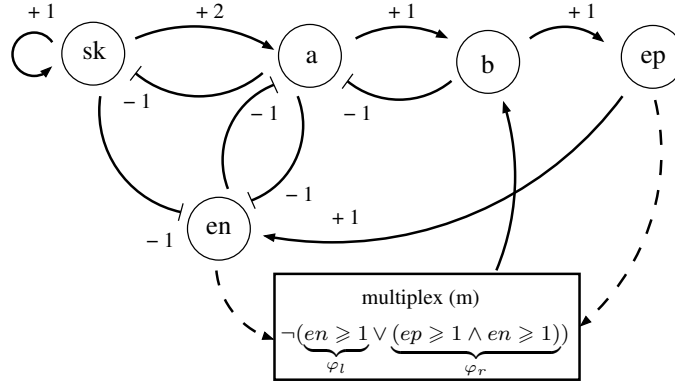


Figure 1.9. Graphe d'influences du cycle cellulaire simplifié des mammifères. Dans cette figure, un arc (u, v) étiqueté par n (resp. $-n$) correspond au multiplexe de formule $(u \geq n)$ (resp. $\neg(u \geq n)$).

Le comportement classique du cycle cellulaire démarre dans l'état ($sk=0, ep=0, a=0, b=0, en=1$) pour revenir à cet état, et passe par 4 phases successives identifiées par des points de synchronisation :

- (G1) la phase de croissance qui prépare la réplication,

- (*S*) la cellule continue de grossir et le centrosome est répliqué,
- (*G2*) la cellule prépare sa division et termine la réplication,
- (*M*) est la phase de mitose pour produire deux cellules sœurs.

1.5.3.1. Comportement temporel

Le comportement temporel global du système est cyclique et peut être modélisé par la formule CTL suivante:

$$AG \left(\left(\begin{array}{l} sk = 0 \wedge ep = 0 \wedge \\ a = 0 \wedge b = 0 \wedge \\ en = 1 \end{array} \right) \implies AX \left(AF \left(\begin{array}{l} sk = 0 \wedge ep = 0 \wedge \\ a = 0 \wedge b = 0 \wedge \\ en = 1 \end{array} \right) \right) \right)$$

qui exprime qu'il est toujours vrai que si l'on est dans l'état initial ($sk=0, ep=0, a=0, b=0, en=1$) alors dans un futur strict (le $AX AF$), on reviendra à cet état.

Il y a au total 100800 paramétrages possibles et *TotemBioNet* sélectionne 13152 paramétrages comme modèles de la formule CTL en 11m01s. Notons qu'il faut ici se servir de la version CTL' de cette formule; en effet, avec le CTL pur, *NuSMV* reste dans un sous-arbre qui correspond à la phase *G0* du cycle cellulaire, et ne cycle pas, tandis qu'en CTL', *NuSMV* revient sur l'état de départ.

1.5.3.2. Trace observée

En s'appuyant sur les connaissances du comportement classique du cycle cellulaire et de ses différentes phases, on peut définir le triplet de Hoare suivant (voir Behaegel *et al.* (2015)):

$$\left\{ \begin{array}{l} sk=0 \wedge \\ ep=0 \wedge \\ a=0 \wedge \\ b=0 \wedge \\ en=1 \end{array} \right\} \underbrace{sk+; sk+; en-; a+; sk-; sk-; b+; a-}_{G1} \underbrace{ep+; en+; b-; ep-}_{M} \underbrace{sk=0 \wedge ep=0 \wedge a=0 \wedge b=0 \wedge en=1}_{S} \left\{ \begin{array}{l} sk=0 \wedge \\ ep=0 \wedge \\ a=0 \wedge \\ b=0 \wedge \\ en=1 \end{array} \right\}$$

où la pré-condition et la post-condition sont l'état sur lequel le système cycle, et les phases *G1*, *S*, *G2*, et *M* sont mises en évidence.

La plus faible pré-condition simplifiée calculée instantanément par *Hoare-fol* est une conjonction d'inégalités. Elle est utilisée dans *TotemBioNet* pour réduire l'espace des paramétrages qui se réduit ainsi à 676 possibilités. 609 de ces paramétrages sont sélectionnés comme modèles valides de la formule CTL en 7s276ms. Cependant, si l'ordre des phases ainsi que les transitions entre les phases sont bien connus, il n'en est pas de même pour l'ordre des transitions à l'intérieur des phases. Un triplet en

meilleure adéquation avec les connaissances biologiques exploite le quantificateur existentiel de la logique de Hoare pour décrire ces différents ordres. La trace devient alors:

```
(sk+;sk+);

Exists((en-;a+;sk-;sk-), (a+;en-;sk-;sk-), (a+;sk-;en-;sk-), (a+;sk-;sk-;en-),
(en-;sk-;a+;sk-), (sk-;en-;a+;sk-), (sk-;a+;en-;sk-), (sk-;a+;sk-;en-),
(en-;sk-;sk-;a+), (sk-;en-;sk-;a+), (sk-;sk-;en-;a+), (sk-;sk-;a+;en-));

Exists((b+;a-), (a-;b+));

Exists((ep+;en+;b-;ep-), (en+;ep+;b-;ep-), (en+;b-;ep+;ep-), (en+;b-;ep-;ep+),
(ep+;b-;en+;ep-), (b-;ep+;en+;ep-), (b-;en+;ep+;ep-), (b-;en+;ep-;ep+),
(ep+;b-;ep-;en+), (b-;ep+;ep-;en+), (b-;ep-;ep+;en+), (b-;ep-;en+;ep+),
(ep+;en+;ep-;b-), (en+;ep+;ep-;b-), (en+;ep-;ep+;b-), (en+;ep-;b-;ep+),
(ep+;ep-;en+;b-), (ep-;ep+;en+;b-), (ep-;en+;ep+;b-), (ep-;en+;b-;ep+),
(ep+;ep-;b-;en+), (ep-;ep+;b-;en+), (ep-;b-;ep+;en+), (ep-;b-;en+;ep+));
```

La plus faible pré-condition simplifiée calculée instantanément par *Hoare-fol* est une *disjonction* de conjonctions qui correspond aux différentes alternatives de traces dans le *Exists*. Cette formule contient 2402 atomes et 515 disjonctions. Elle est utilisée lors de l'énumération pour filtrer les paramétrages: les automates construits et testés par *NuSMV* sont ceux qui satisfont cette formule. Parmi les 100800 paramétrages, 99944 ne satisfont pas cette formule et parmi les 856 restants, *TotemBioNet* en sélectionne 777 en 9s451ms. On peut vérifier que naturellement, ce nombre de paramétrages est supérieur à celui qui a été trouvé pour la trace particulière précédente. D'autre part, bien que la trace soit plus générale, le temps d'exécution est sensiblement identique.

1.6. Formalisme hybride

Dans un grand nombre de phénomènes biologiques, la composante temporelle joue un rôle primordial, permettant à des influences de se produire à des moments clés d'autres processus. Par exemple, l'horloge circadienne permet à chaque cellule de s'adapter à l'alternance jour / nuit, et d'anticiper les changements d'environnement quotidiens. Pour raisonner sur de tels systèmes, il devient nécessaire d'avoir une représentation du temps passé dans chacun des états. Le cadre de modélisation choisi ici consiste à considérer que les vitesses de synthèse et/ou dégradation des différentes espèces chimiques dans chacun des états qualitatifs sont constantes.

Pour illustrer ce cadre de modélisation hybride, nous étendrons l'exemple de l'horloge circadienne HCSM présenté en section 1.2, figure 1.5.

1.6.1. Le formalisme hybride

Le formalisme reprend une grande partie du formalisme purement qualitatif puisqu'il s'agit d'ajouter une composante temporelle au cadre qualitatif. Nous retrouvons donc dans la définition suivante les variables, multiplexes et arcs. En revanche, pour prendre en compte des vitesses de synthèse/dégradation, nous remplaçons les paramètres $K_{v,\omega}$ du cadre de modélisation qualitatif par des *célérités* $C_{v,\omega,n}$ dépendant non seulement de la variable considérée et de ses ressources mais aussi de son niveau d'expression qualitatif.

DEFINITION 1.11 (Réseau de Régulation Hybride (RRH)). – *Un réseau de régulation hybride (RRH) est un n -uplet $R = (\Sigma = (V \cup M, A), \mathcal{C}, \sigma)$ où :*

- V est des variables v de domaines discrets $\llbracket 0, b_v \rrbracket$,
- M , l'ensemble des multiplexes m de formules associées φ_m , où les φ_m sont des formules de multiplexes de $(V \cup M)$,
- A est l'ensemble des arcs de la forme $(m \rightarrow v) \in M \times V$.
- $\mathcal{C} = \{C_{v,\omega,n}\}$ est une famille de paramètres appelés *célérités* indexée par un triplet (v, ω, n) où $v \in V$, ω est un sous ensemble de $\Sigma^{-1}(v)$, et $n \in \llbracket 0, b_v \rrbracket$. $C_{v,\omega,n}$ est appelé la *célérité* de v pour ω au niveau n .
- $\sigma : \mathcal{C} \rightarrow \mathbb{R}$ est une interprétation des symboles de $\mathcal{C}$ dans l'ensemble des réels. Elle doit satisfaire les contraintes suivantes:

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

EXEMPLE 1.8. – HCSM. Bien évidemment, les variables G et PC restent présentes. Cependant, pour modéliser l'alternance du jour et de la nuit, une seule variable L n'est plus suffisante car on verra que la sémantique hybride ne permet pas des trajectoires cycliques lorsqu'il n'y a qu'une seule variable. On ajoute donc une variable X permettant de contrôler les durées du jour et de la nuit, voir figure 1.10. Les éléments de V (ensemble de variables) sont donc $\{G, PC, L, X\}$, ceux de M (ensemble des multiplexes) sont $\{m_1, m_2, m_3, m_4\}$ et l'ensemble A (ensemble d'arcs) se déduit de la figure. La famille de symboles associés à ce modèle est maintenant la famille des *célérités* $\mathcal{C} = \{C_{v,\omega,n}\}$ où $v \in \{G, PC, L, X\}$, ω est un ensemble de ressources et n un niveau d'expression qualitatif. Comme chacune des 4 variables est binaire, que chacune d'entre elles a un et un seul multiplexe régulateur, le modèle hybride contient 16 paramètres énumérés dans la partie droite de la figure, qui donne aussi une interprétation possible σ .

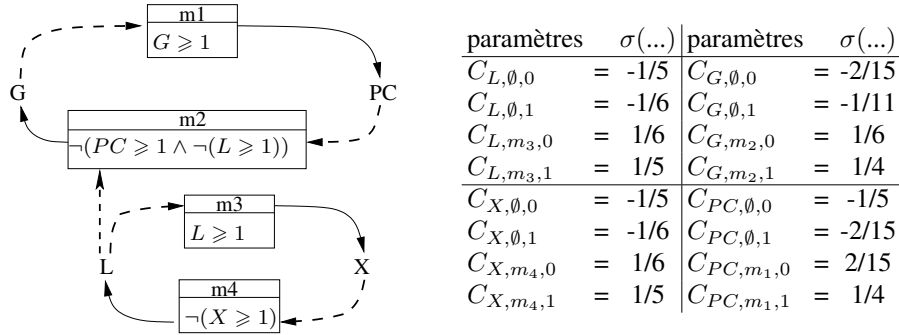


Figure 1.10. RRH pour l'horloge circadienne. A droite, l'énumération de tous les paramètres avec un exemple d'interprétation possible.

1.6.2. Définition des trajectoires hybrides

Intuitivement, les trajectoires hybrides sont maintenant des suites de trajectoires rectilignes à vitesse constante à l'intérieur de chaque état qualitatif, intercalées avec des transitions entre états qualitatifs. Pour définir cette nouvelle sémantique, nous commençons par introduire les états d'un RRH qui ne sont que la donnée d'un état qualitatif et d'une partie dite *fractionnaire* représentant la position exacte du système à l'intérieur de l'état qualitatif.

DEFINITION 1.12 (État d'un RRH).— Soit $R = ((V \cup M, A), \mathcal{C}, \sigma)$ un RRH. Un état hybride de R est un couple $h = (\eta, \pi)$ où

- η est une interprétation qui associe à chaque variable v de V une valeur dans $\llbracket 0, b_v \rrbracket$,
- π est une autre interprétation qui associe à chaque variable v de V une valeur dans l'intervalle réel $[0, 1]$.

η est appelé l'état discret ou état qualitatif de h et π est appelé sa partie fractionnaire. Par abus de notation, on notera par la suite $\eta_v = \eta(v)$ et $\pi_v = \pi(v)$. Nous dénotons S l'ensemble des états hybrides de R . Lorsqu'il n'y a pas d'ambiguïté, on utilisera les notations η et π sans mention explicite du couple h .

EXEMPLE 1.9.— HCSM. La figure 1.11 (gauche) reprend uniquement le sous réseau de régulation hybride qui représente l'alternance de la lumière pour l'étude de l'HCSM. Ce sous-réseau peut être étudié de manière indépendante, car il contrôle les trajectoires de G et PC mais n'est contrôlé par aucune régulation externe. Les variables L et X sont binaires, on en déduit donc qu'il y a quatre états discrets, et chaque état hybride est caractérisé par un état qualitatif et un "vecteur" représentant la position précise à l'intérieur de l'état qualitatif.

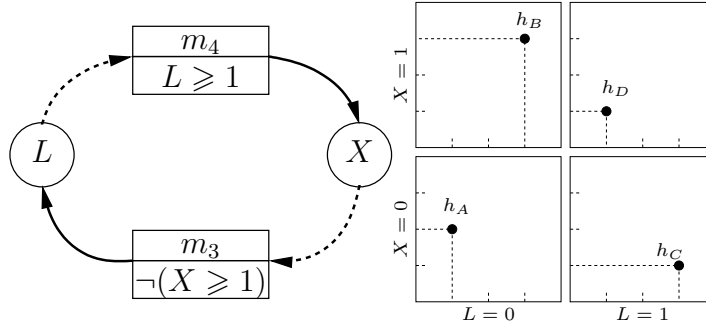


Figure 1.11. Gauche: le sous réseau de régulation permettant l'alternance de la lumière. Droite: illustration des notions d'état qualitatif et de partie fractionnaire: L'état h_A est un état hybride dont l'état discret est noté par abus de notation $\eta = (0, 0)^t$ et la partie fractionnaire est notée $\pi = (0.25, 0.5)^t$. On peut donc écrire $h_a = \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.25 \\ 0.5 \end{pmatrix} \right)$.

Puisque les formules des multiplexes ne dépendent pas des parties fractionnaires de l'état hybride, la notion de ressource reste inchangée et ne dépend que de l'état discret considéré.

EXEMPLE 1.10.– HCSM. En reprenant la figure 1.11-droite, on peut calculer les ressources de L et X en chacun des états hybrides h_A , h_B , h_C et h_D :

	en h_A	en h_B	en h_C	en h_D
ressource de L :	$\{m_3\}$	$\{\}$	$\{m_3\}$	$\{\}$
ressource de X :	$\{\}$	$\{\}$	$\{m_4\}$	$\{m_4\}$

Le paramètre $C_{v,\omega,n}$ représente la vitesse d'évolution de v lorsque v est soumis à l'ensemble de ressources ω et que le niveau d'expression qualitatif de v est n . De fait, l'indice supplémentaire (n) permet de choisir des valeurs distinctes de célérité de v dans différents états qualitatifs même lorsque ces états ont mêmes ressources pour v , voir la figure 1.12. Notez que ces célérités $C_{v,\omega,n}$ ne dépendent pas de la partie fractionnaire de l'état hybride courant.

On étend donc la notion de *paramètre applicable* aux célérités en prenant en compte l'état qualitatif de v : $C_{v,\omega,n}$ est *applicable* pour $h = (\eta, \pi)$ si ω est l'ensemble de ressources de v pour η et si $\eta_v = n$.

La figure 1.13 matérialise les célérités dans chacun des 4 états discrets du sous-réseau considéré dans la figure 1.11. On peut maintenant chercher à calculer le temps que le système va mettre pour sortir de l'état qualitatif courant, et pour cela, nous



Figure 1.12. La célérité de v peut ne pas être la même lorsqu'on compare plusieurs états qualitatifs ayant les mêmes ressources. Si on suppose que v a, dans les quatre états de la figure, le même ensemble ω de ressources, on peut choisir les célérités de telle manière que la célérité diminue au fur et à mesure qu'on se rapproche de son point focal. Cela peut par exemple représenter un modèle différentiel linéaire dans lequel la vitesse d'évolution diminue au fur et à mesure que le système se rapproche du point d'équilibre.

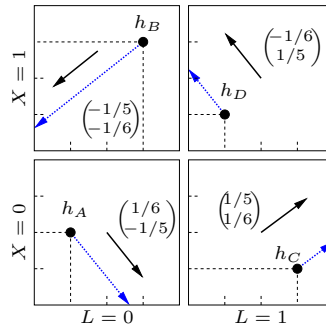


Figure 1.13. A chaque état discret est associé un vecteur de célérités (vecteurs noirs). Les valeurs utilisées dans cette figure, sont celles de la figure 1.10. Les flèches bleues en pointillés représentent les trajectoires en partant de chacun des états hybrides h_A , h_B , h_C et h_D jusqu'à ce qu'elles rencontrent un bord de l'état qualitatif courant.

introduisons le *délai de contact* de chacune des variables: pour une variable v et un état hybride $h = (\eta, \pi)$, le *délai de contact* $\delta_h(v)$ de v en h représente le temps nécessaire à v pour atteindre le bord de l'état qualitatif. En notant ω l'ensemble des ressources de v en h , le délai se calcule comme suit:

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

Dans la figure 1.13, en partant de l'état hybride h_A , la trajectoire va toucher le bord “sud” de l'état qualitatif. En supposant que les célérités sont exprimées en unité/h, le temps nécessaire pour toucher ce bord est égal à 2,5h car la célérité $C_{X,\emptyset,0}$

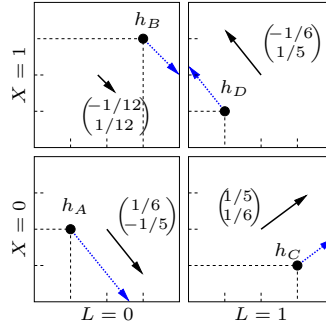


Figure 1.14. Le vecteur de célérités de l'état qualitatif $(0, 1)$ a été modifié par rapport à la figure 1.13. Dans cette situation, la trajectoire partant de l'état hybride h_D atteint son bord, mais ne peut pas passer dans l'état discret $(0, 1)$ parce que dans ce nouvel état, les trajectoires s'opposent à l'entrée de trajectoires provenant de l'état discret $(1, 1)$.

applicable dans cet état vaut $-1/5$, et la variable X doit parcourir la moitié d'une unité. Pour toucher l'autre bord ($\pi_L = 1$), il aurait fallu attendre 4,5h car la célérité de L applicable vaut $1/6$ et il faut parcourir $3/4$ d'une unité.

Une fois que la trajectoire a atteint le bord le plus proche dans l'état qualitatif, il n'est pas encore certain qu'elle va sortir de l'état qualitatif courant. Deux situations différentes peuvent avoir lieu pour empêcher la transition vers un autre état qualitatif: soit il n'existe pas d'état qualitatif de l'autre côté du mur (c'est le cas des trajectoires commençant en h_A , h_B et h_C de la figure 1.13), soit il en existe bien un, mais les vitesses dans ce nouvel état qualitatif sont opposées, de sorte qu'il n'est pas possible de franchir le seuil entre les deux états. Cette deuxième situation est illustrée dans la figure 1.14.

Ces remarques permettent d'introduire la notion de murs externes et internes. Nous dirons que v fait face à un mur externe en h si la célérité de v l'emmène en direction de l'extérieur de son domaine de définition, autrement dit si:

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

En h_A de la figure 1.13, la variable X fait face à un mur externe parce que la célérité applicable est négative et que $\eta_X = 0$, et en h_C , la variable L fait face à un mur externe parce que la célérité applicable est positive et que $\eta_L = b_L$.

Nous dirons que la variable v fait face à un mur interne en h lorsque la seconde situation se produit, autrement si $\text{signe}(C_{v,\omega,\eta_v}) \times \text{signe}(C_{v,\omega',\eta'_v}) = -1$, où ω est l'ensemble des ressources de v en h , où η' est l'état discret de l'autre côté du seuil et où ω' est l'ensemble des ressources de v en η' . On notera $sv(h)$ l'ensemble des

variables *glissantes* en h , c'est à dire des variables qui font face à un mur externe ou interne en h .

Lorsqu'on regarde la trajectoire en partant de l'état hybride h_A de la figure 1.13, la variable X fait face à un mur externe, L ne fait pas face à un mur, puisque les célérités de L en $(0, 0)$ et en $(1, 0)$ sont toutes les deux de même signe (positives). On en déduit que $sv(h_A) = \{X\}$. Pour l'état qualitatif $(1, 1)$ de la figure 1.14, la variable X fait face à un mur externe et L fait face à un mur interne. On en déduit que $sv(h_D) = \{L, X\}$.

On peut maintenant rechercher les variables susceptibles de changer de valeur qualitative.

DEFINITION 1.13 (Variables changeant en premier).— Soient $R = ((V \cup M, A), \mathcal{C}, \sigma)$ un RRH et $h = (\eta, \pi)$ un état hybride. L'ensemble des variables changeant en premier est défini par:

$\text{first}(h) = \{v \in V \setminus sv(h) \mid \delta_h(v) \neq +\infty \wedge \forall u \in V \setminus sv(h), \delta_h(u) \geq \delta_h(v)\}$.
De plus, δ_h^{first} dénote le temps passé dans l'état qualitatif de h lorsqu'on part de l'état h : pour tout $v \in \text{first}(h)$, $\delta_h^{\text{first}} = \delta_h(v)$, ou bien $\delta_h^{\text{first}} = +\infty$ si $\text{first}(h) = \emptyset$.

Pour l'état hybride h_A de la figure 1.13, on a $\text{first}(h_A) = \{L\}$ et $\delta_{h_A}^{\text{first}} = 4, 5h$. A l'opposé, pour l'état hybride h_D de la figure 1.14, on a $\text{first}(h_D) = \{\}$ et $\delta_{h_D}^{\text{first}} = \infty$.

L'espace de transitions hybrides peut maintenant être défini.

DEFINITION 1.14 (Espace de transitions hybrides).— Soit $R = ((V \cup M, A), \mathcal{C}, \sigma)$ un RRH. On note $\mathcal{R} = (S, cT, dT)$ l'espace de transitions hybrides de R où S est l'ensemble des états hybrides, et cT (resp. dT) l'ensemble des transitions continues (resp. discrètes):

Il existe une transition continue dans cT de l'état $h = (\eta, \pi)$ vers l'état $h' = (\eta', \pi')$ ssi:

- 1) soit $\text{first}(h) \neq \emptyset$ et il existe une variable $v \in \text{first}(h)$ telle que:
 - a) $\delta_h(v) \neq 0$, où $\delta_h(v)$ est appelé la durée de la transition (continue),
 - b) $\eta' = \eta$ et $\pi'_u = \begin{cases} 0 & \text{si la célérité applicable } C_{u,\dots} < 0 \\ 1 & \text{si la célérité applicable } C_{u,\dots} > 0 \end{cases}$ pour tout $u \in (\text{first}(h) \cup \nu)$ où $\nu = \{x \in sv(h) \mid \delta_h(x) \leq \delta_h^{\text{first}}\}$,
 - c) pour tout $z \in V \setminus (\text{first}(h) \cup \nu)$, on a $\pi'_z = \pi_z + \delta_h(v) \times C_{z,\dots}$ où $C_{z,\dots}$ est la célérité applicable pour z .

2) soit $\text{first}(h) = \emptyset$ (autrement dit, chaque variable v est soit dans un état où la célérité applicable $C_{v,\dots}$ est nulle, soit dans un état où elle fait face à un mur, $v \in sv(h)$) et:

- a) $\forall v \in sv(h), \pi'_v = \begin{cases} 0 & \text{si la célérité applicable } C_{v,\dots} < 0 \\ 1 & \text{si la célérité applicable } C_{v,\dots} > 0 \end{cases}$

$b) \forall u \notin \text{sv}(h), \pi'_u = \pi_u$ (puisque dans ce cas la célérité applicable $C_{u,\dots}$ est nulle).

Il existe une transition discrète dans dT de l'état $h' = (\eta', \pi')$ vers l'état $h'' = (\eta'', \pi'')$ ssi il existe une variable $v \in \text{first}(h')$ telle que:

- 1) $\delta_{h'}(v) = 0$, où $\delta_{h'}(v)$ est appelé la durée de la transition discrète,
- 2) $\eta''_v = \eta'_v + \text{signe}(C_{v,\dots})$ et $\pi''_v = \begin{cases} 0 & \text{si la célérité applicable } C_{v,\dots} > 0 \\ 1 & \text{si la célérité applicable } C_{v,\dots} < 0 \end{cases}$,
- 3) $\forall u \in V \setminus \{v\}, \eta''_u = \eta'_u$ et $\pi''_u = \pi'_u$.

Les états à partir desquels il n'existe aucune transition (discrète ou continue) sont appelés **états stables**.

L'intérêt de cette formalisation réside dans la possibilité de représenter formellement les durées d'attente dans chacun des états. Il faut cependant attirer l'attention sur le fait que l'hypothèse d'une célérité uniforme à l'intérieur d'un état qualitatif peut mener à construire des modèles ayant certains biais, comme l'illustre la figure 1.15. Si l'on souhaite représenter une oscillation entretenue, il faut un jeu de célérités qui conduit les trajectoires à toucher au moins un des bords, et ainsi, le modèle exhibe des trajectoires convergeant en un temps fini vers la trajectoire cyclique, c'est le cas de la figure 1.16.

1.7. Logique de Hoare Hybride

On peut munir les modèles hybrides d'une logique de Hoare afin de pouvoir vérifier si un modèle admet une trace continue représentée par un triplet de Hoare. De même que dans le cas discret, on produit un ensemble de contraintes qui caractérise l'ensemble des interprétations (des célérités) qui respectent ce triplet. L'idée est similaire à la section 1.3, mais la nature continue de la trace ainsi que le caractère quasi déterministe des modèles hybrides font émerger certaines difficultés.

Le langage de traces doit bien évidemment décrire les changements qualitatifs (passages de seuils avec des expressions du type $v+$ ou $v-$) et spécifier le temps passé dans chacun des états qualitatifs, mais aussi encoder des observations quantitatives, comme la dégradation totale ou au contraire la saturation d'une variable. Ainsi, les atomes de chemins hybrides seront des triplets représentant le temps passé dans l'état qualitatif présent, la prochaine transition discrète ainsi que des informations complémentaires sur les comportements aux bords (rencontre de mur extérieur ou infranchissable, et donc *glissement* des variables ainsi bloquées pendant que les autres variables évoluent).

Par ailleurs, la majorité des trajectoires d'un modèle hybride sont déterministes, et les traces temporisées observées ne sont pas nombreuses. Nous n'exposerons donc

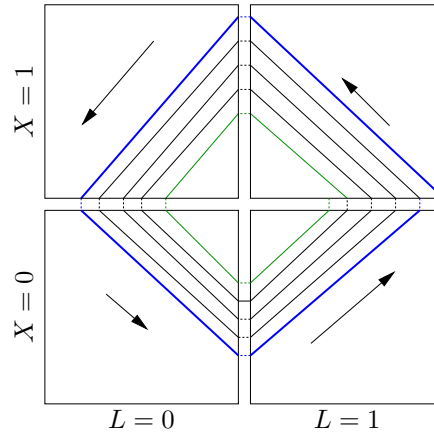


Figure 1.15. Dans un modèle 2D d'une oscillation (ici le système L - X introduit pour représenter l'oscillation de la lumière), en fonction des célérités choisies, les trajectoires peuvent toutes converger vers le centre. Puisque les célérités sont constantes à l'intérieur de chaque domaine, le temps nécessaire pour revenir à un même état qualitatif diminue et tend même vers 0 : parcourir la partie verte de la trajectoire demande beaucoup moins de temps que parcourir la partie bleue. Ainsi la période d'oscillation diminue autant qu'on le souhaite et la vitesse angulaire augmente jusqu'à l'infini.

pas ici le traitement des quantificateurs, ni les conditionnelles ou les boucles, d'utilité moindre que dans le cas purement qualitatif.

On peut alors définir des règles d'inférences et le calcul de plus faible précondition associé qui assurent au modèle d'exhiber la trace spécifiée et d'arriver à un état qui satisfait la post-condition. La logique ainsi construite est correcte et complète sémantiquement (i.e. on calcule bien la plus faible précondition).

1.7.1. Langages de propriétés, de chemins et d'assertion

Cette section passe en revue les différents éléments permettant d'écrire un triplet de Hoare. Nous commençons par décrire le langage de propriétés dans lequel les pré- et post-conditions sont écrites. Puisque un état est décrit par un état qualitatif et une position précise à l'intérieur, toute expression du langage de propriété sera découpée en deux parties: l'une parlant exclusivement des états qualitatifs, et l'autre faisant le lien entre les positions précises à l'intérieur des états qualitatifs et la trace discrète.

Les termes du langage de propriétés sont inductivement définis comme suit:

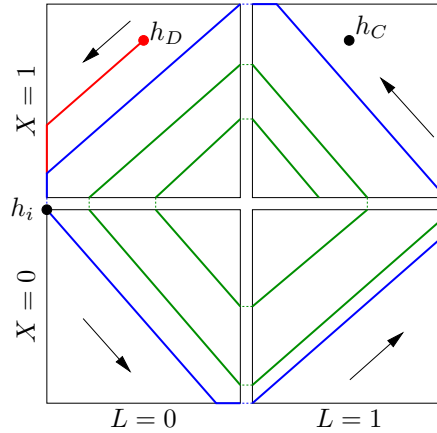


Figure 1.16. Les célérités de L et X ont été choisies (exemple 1.8) pour qu'il existe une unique trajectoire cyclique (trajectoire bleue). Les célérités égales à $1/6$ (exprimé en unité par heure) permettent de rester dans chacun des états qualitatifs $6h$, et les célérités égales à $\pm 1/5$ assurent que dans chaque carré, on va toucher le mur externe. La trajectoire cyclique est alors parcourue en exactement $24h$. Lorsqu'on part de l'extérieur (trajectoire rouge), en moins d'un tour, on se retrouve sur la trajectoire cyclique. Lorsqu'on part de l'intérieur (trajectoire verte), il faut plusieurs tours pour atteindre la trajectoire cyclique.

- Un *terme discret* est une variable η_v avec $v \in V$, ou une constante de $\mathbb{N}$;
- Un *terme continu* est une variable π_v ou π'_v avec $v \in V$, ou une célérité $C_{v,\omega,n}$ avec $v \in V$, $\omega \subset \Sigma^{-1}(v)$ et $n \in \llbracket 0, b_v \rrbracket$, ou une constante de $\mathbb{R}$;
- Les opérateurs $+$, $-$, $\times$ et $/$ créent des nouveaux termes par composition, le dernier étant seulement défini pour les termes continus. Nous utilisons leurs sémantiques usuelles.

Les *atomes discrets* (resp. *atomes continus*) sont de la forme $n \square n'$ où n et n' sont des termes discrets (resp. $f \square f'$ où f et f' sont des termes continus) et où $\square$ est un comparateur ($\leq$, $\geq$, $=$, $<$ ou $>$).

Dès lors, les *conditions discrètes* sont définies par: $D := a_d | \neg D | D \wedge D | D \vee D$ et les *conditions hybrides* sont définies par: $H := a_d | a_c | \neg H | H \wedge H | H \vee H$ où a_d et a_c sont respectivement un atome discret et un atome continu.

Enfin 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 $\mathcal{L}_{\mathcal{P}}$.

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

Le lecteur averti remarquera que, dans cette définition, le langage des conditions discrètes est inclus dans celui des conditions hybrides. Cependant d'un point de vue méthodologique il est préférable de regrouper les conditions qui ne traitent que des états qualitatifs. Les conditions discrètes reprennent les conditions de la logique de Hoare discrète présentée en section 1.3, alors que les conditions hybrides vont permettre de coder des informations sur le comportement à l'intérieur d'un état qualitatif.

EXEMPLE 1.11. – HCSM. En reprenant la figure 1.16, on peut vouloir dire qu'au départ de la trajectoire rouge, le système se trouve dans l'état qualitatif ($L = 0, X = 1$) et plus précisément que la partie fractionnaire de l'état initial est ($\pi_L = 0,5, \pi_X = 0,8$). On posera alors $D = (\eta_L = 0 \wedge \eta_X = 1)$ et $H = (\pi_L = 0,5 \wedge \pi_X = 0,8)$ et $\phi = (D, H)$. Bien évidemment, on a $h_D \models \phi$ alors que $h_C \not\models \phi$ bien que la partie fractionnaire de h_C soit la même que celle de h_D .

La définition du langage des chemins repose sur deux éléments constitutifs que sont les *atomes de chemin discret* permettant de décrire pas à pas une trace observée qualitative et les *assertions* permettant de contraindre des célérités ou décrire les glissements (ou non glissements) éventuels de variables.

– Les atomes de chemin discret sont les mêmes que dans le cas discret: un changement élémentaire peut donc s'écrire simplement $v+$ si la variable v augmente, ou $v-$ si elle diminue. Pour tous les états $h = (\eta, \pi)$ et $h' = (\eta', \pi')$, la transition $h \xrightarrow{v+} h'$ (resp. $h \xrightarrow{v-} h'$) sera dite satisfaite si et seulement si il existe une transition discrète de h vers h' telle que $\eta'_v = \eta_v + 1$ (resp. $\eta'_v = \eta_v - 1$).

– Le langage d'assertion va permettre de spécifier des contraintes portant soit sur une célérité, soit sur l'existence ou la non-existence de glissement sur un bord de l'état qualitatif courant. On définit alors le langage d'assertion $\mathcal{L}_A$ 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 $v \in V$, $\square$ représente les comparateurs classiques et $c \in \mathbb{R}$. 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.

Intuitivement, une assertion du type $C_v \square c$ exprime une contrainte sur la célérité de la variable v dans l'état qualitatif courant; une assertion du type $\text{slide}^+(v)$ (resp. $\text{slide}^-(v)$) signifie que la variable v atteint son seuil supérieur (resp. inférieur) sans pour autant franchir le seuil; $\text{slide}(v)$ impose le glissement de la variable v soit sur le bord supérieur soit sur le bord inférieur; $\text{noslide}^+(v)$ et $\text{noslide}^-(v)$ indiquent

l'absence de glissement et $noslide(v)$ impose l'absence de glissement sur chacun des deux bords.

EXAMPLE 1.12. – *HCSM*. En reprenant la figure 1.16, on peut vouloir dire qu'en partant de h_D , point de départ de la trajectoire rouge, la célérité applicable de X est inférieur à $-1/10$, que la variable L va atteindre son minimum avant même le premier changement d'état qualitatif, que la variable X ne va pas glisser sur un de ses seuils dans l'état qualitatif courant, et que le prochain changement d'état qualitatif aura lieu dans $3h$. Ces informations se transcrivent simplement par le couple d'assertion:

$$(3h, (C_X \leq -\frac{1}{10}) \wedge slide^-(L) \wedge noslide(X))$$

Pour vérifier qu'une assertion est bien satisfaite à un état hybride donné, il est nécessaire de faire le lien entre les transitions continues (à l'intérieur d'un même état qualitatif) et ces assertions. C'est le but de la définition suivante.

DEFINITION 1.15 (Sémantique du couple d'assertion $(\Delta t, a)$). – *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 $h \rightarrow h'$ 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 $\delta_h^{first} = \Delta t$.
- Si a est de la forme $(C_u \square c)$, $(h, h') \models (\Delta t, a)$ si et seulement si $\delta_h^{first} = \Delta t$ et $(C_{u, \omega, \eta_u} \square c)$ où ω est l'ensemble des ressources de u dans η .
- Si a est de la forme $slide(v)$, $(h, h') \models (\Delta t, a)$ ssi $\delta_h^{first} = \Delta t$ et $\delta_h(v) < \delta_h^{first}$.
- Si a est de la forme $slide^+(v)$ (resp. $slide^-(v)$), $(h, h') \models (\Delta t, a)$ ssi $\delta_h^{first} = \Delta t$ et $\delta_h(v) < \delta_h^{first}$ et $C_{v, \omega, \eta_v} > 0$ (resp. $C_{v, \omega, \eta_v} < 0$) où ω est l'ensemble des ressources de v dans η .
- Si a est de la forme $noslide(v)$, $(h, h') \models (\Delta t, a)$ ssi $\delta_h^{first} = \Delta t$ et $\delta_h(v) \geq \delta_h^{first}$.
- Si a est de la forme $noslide^+(v)$ (resp. $noslide^-(v)$), $(h, h') \models (\Delta t, a)$ ssi $\delta_h^{first} = \Delta t$ et $(C_{v, \omega, \eta_v} > 0) \Rightarrow (\delta_h(v) \geq \delta_h^{first})$ (resp. $(C_{v, \omega, \eta_v} < 0) \Rightarrow (\delta_h(v) \geq \delta_h^{first})$).
- Si a est de la forme $\neg a'$, $(h, h') \models (\Delta t, a)$ si et seulement si $\delta_h^{first} = \Delta t$ 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) $(h, h') \models (\Delta t, a'')$.

Les chemins peuvent maintenant être définis comme une suite de triplets $(\delta t, a, v \pm)$ grâce au constructeur de composition de chemin “;”. La grammaire des chemins est donnée 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 $h_3 \xrightarrow{p_2} h_2$

EXEMPLE 1.13. – *HCSM. Pour spécifier que le sous système L - X de la figure 1.11 présente une oscillation de $24h$, durant laquelle, le système passe dans chacun des 4 états exactement $6h$ avec des glissement dans chacun d'eux, on peut écrire le chemin suivant (qui suppose un état initial dans l'état qualitatif ($L = 0, X = 0$)):*

$$\left(\begin{array}{c} 6h, \\ \text{slide}^-(X), \\ L+ \end{array} \right); \left(\begin{array}{c} 6h, \\ \text{slide}^+(L), \\ X+ \end{array} \right); \left(\begin{array}{c} 6h, \\ \text{slide}^+(X), \\ L- \end{array} \right); \left(\begin{array}{c} 6h, \\ \text{slide}^-(L), \\ X- \end{array} \right)$$

Nous voyons bien ici que les atomes de chemin ne vont pas permettre d'être très expressif pour les chemins. Un chemin est finalement une séquence finie de chemins élémentaires et chaque chemin élémentaire décrit la future transition discrète qui va avoir lieu, le temps à passer dans l'état qualitatif courant avant la transition discrète, ainsi que certaines informations sur la transition continue.

1.7.2. Triplets de Hoare

Un triplet de Hoare hybride pour un réseau de régulation hybride donné est donc simplement une expression de la forme $\{Pre\}p\{Post\}$ où $Pre \in \mathcal{L}_{\mathcal{P}}$ est la précondition, $Post \in \mathcal{L}_{\mathcal{P}}$ la postcondition, et p est un chemin. 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 $h_2 \models Post$.

EXEMPLE 1.14. – *HCSM. En reprenant le chemin exhibé précédemment, voir exemple 1.13, on pourrait vouloir spécifier qu'après $24h$, on se retrouve exactement au même état hybride, autrement dit qu'il existe un cycle de $24h$ dans la dynamique. Ici, puisqu'il y a glissement dans chacun des états qualitatifs, l'état hybride initial peut être totalement caractérisé: c'est le seul état hybride h_i tel que $\eta_{h_i} = (0, 0)^t$ et $\pi_{h_i} = (0, 0, 1, 0)^t$, voir figure 1.16. Le triplet de Hoare reprenant ces spécifications*

s'écrit alors :

$$\left(\begin{array}{c} 6h, \\ slide^-(X), \\ L+ \end{array} \right); \left(\begin{array}{c} 6h, \\ slide^+(L), \\ X+ \end{array} \right); \left(\begin{array}{c} 6h, \\ slide^+(X), \\ L- \end{array} \right); \left(\begin{array}{c} 6h, \\ slide^-(L), \\ X- \end{array} \right)$$

$$\left\{ \left(\begin{array}{c} D_{pre} = (\eta_L = 0 \wedge \eta_X = 0), \\ H_{pre} = (\pi_L = 0.0 \wedge \pi_X = 1.0) \end{array} \right) \right\}$$

$$\left\{ \left(\begin{array}{c} D_{post} = (\eta_L = 0 \wedge \eta_X = 0), \\ H_{post} = (\pi_L = 0.0 \wedge \pi_X = 1.0) \end{array} \right) \right\}$$

1.7.3. Calcul de la plus faible précondition

Lorsqu'on se donne une post-condition ainsi qu'un chemin, nous pouvons, comme nous l'avons déjà fait pour les modèles purement qualitatifs en section 1.3, définir inductivement la plus faible précondition sur l'état de départ et sur les paramètres du modèle, qui permettent au modèle d'exhiber le chemin donné et arriver à un état satisfaisant la post-condition.

DEFINITION 1.16 (Plus faible précondition).— 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 $p = \varepsilon$, alors $D_{i,f} \equiv D_f$ et $H_{i,f} \equiv H_f$;
- Si $p = (\Delta t, a, v+)$, alors

$$\begin{cases} D_{i,f} \equiv D_f[\eta_v \setminus \eta_v + 1], \\ H_{i,f} \equiv H_f[\eta_v \setminus \eta_v + 1] \wedge \Phi_v^+(\Delta t) \wedge \mathcal{F}(\Delta t) \wedge \neg \mathcal{W}_v^+ \wedge \mathcal{A}(\Delta t, a) \wedge \mathcal{J}_v; \end{cases}$$
- Si $p = (\Delta t, a, v-)$, alors

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

où les propriétés $\Phi_v^+(\Delta t)$, $\Phi_v^-(\Delta t)$, $\mathcal{W}_v^+$, $\mathcal{W}_v^-$, $\mathcal{F}(\Delta t)$, $\mathcal{A}(\Delta t, a)$ and $\mathcal{J}_v$ sont des sous propriétés détaillés dans Behaegel (2018).

Intuitivement, chacune des sous-propriétés décrit une condition qui doit être satisfaite pour permettre les transitions continue et discrète associées à un chemin élémentaire:

- $\Phi_v^+(\Delta t)$ (resp. $\Phi_v^-(\Delta t)$) correspond à la contrainte permettant à l'entité v d'atteindre son bord supérieur (resp. inférieur) en un temps Δt et assurant la transition discrète;
- $\mathcal{F}(\Delta t)$ contraint toutes les entités sauf l'entité v du chemin élémentaire à atteindre leurs bords après v , ou à faire face à un mur;
- $\neg \mathcal{W}_v^\pm$ interdit la présence d'un mur pour l'entité v , l'autorisant ainsi à changer de niveau qualitatif;
- $\mathcal{A}(\Delta t, a)$ traduit les formules de l'assertion a en contraintes spécifiques à l'état courant;
- $\mathcal{J}_v$ fait la jonction entre deux états successifs.

1.7.4. Règles d'inférence

De la même manière qu'en section 1.3, on associe à chaque construction syntaxique du chemin, une *règle d'inférence*. Ici on va donc avoir trois règles d'inférence: incrémentation, décrémentation, et composition séquentielle.

1) Règle d'incrémentation :
$$\frac{}{\left\{ \begin{array}{c} D_f[\eta_v \leftarrow \eta_v + 1] \\ H_{i,f} \end{array} \right\} \left(\begin{array}{c} \Delta t \\ a \\ v+ \end{array} \right) \left\{ \begin{array}{c} D_f \\ H_f \end{array} \right\}}$$

2) Règle de décrémentation :
$$\frac{}{\left\{ \begin{array}{c} D_f[\eta_v \leftarrow \eta_v - 1] \\ H_{i,f} \end{array} \right\} \left(\begin{array}{c} \Delta t \\ a \\ v- \end{array} \right) \left\{ \begin{array}{c} D \\ H_f \end{array} \right\}}$$

où v est une variable, η_v son niveau qualitatif, D_f la condition discrète, H_f la condition hybride, $H_{i,f} \equiv H_f[\eta_v \leftarrow \eta_v + 1] \wedge \Phi_v^-(\Delta t) \wedge \mathcal{F}(\Delta t) \wedge \neg \mathcal{W}_v^- \wedge \mathcal{A}(\Delta t, a) \wedge \mathcal{J}_v$; (resp. $H_{i,f} \equiv H_f[\eta_v \leftarrow \eta_v - 1] \wedge \Phi_v^-(\Delta t) \wedge \mathcal{F}(\Delta t) \wedge \neg \mathcal{W}_v^- \wedge \mathcal{A}(\Delta t, a) \wedge \mathcal{J}_v$) la condition hybride d'incrémentation (resp. de décrémentation), Δt le temps passé dans l'état qualitatif courant et a une assertion.

3) Règle de composition séquentielle :
$$\frac{\{Q_1\} p_1 \{Q_3\}}{\{Q_1\} p_1; p_2 \{Q_2\}} \frac{\{Q_3\} p_2 \{Q_2\}}{\{Q_1\} p_1; p_2 \{Q_2\}} \text{ (classique donc).}$$

Trois schémas d'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,

– $\forall v \in V, \forall \omega \subset \Sigma^{-1}(v), \forall k \in \llbracket 0, b_v - 1 \rrbracket, C_{v,\omega,k} \times C_{v,\omega,k+1} \geq 0$: pour deux états qualitatifs voisins, si l'entité v est contrôlée par le même ensemble de ressources, alors les célérités de v ne peuvent être de signes opposés.

Ce second schéma d'axiomes impose que les célérités d'une entité v dans deux états voisins qui donnent les mêmes ressources à v , ne puissent pas faire évoluer la variable

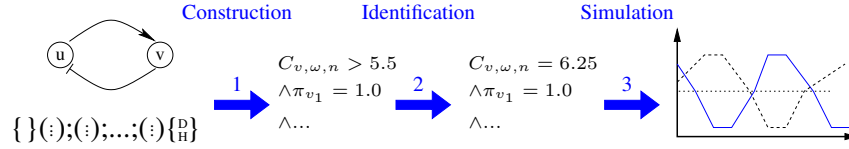


Figure 1.17. Les 3 étapes du prototype Holmes BioNet sont indiquées en bleu et correspondent à la construction des contraintes sur les paramètres dynamiques (1), l'identification d'un ou plusieurs jeux de paramètres (2) et la simulation de la dynamique à partir d'un jeu de paramètres choisi (3).

v dans des directions opposées, ce qui semble raisonnable puisque les ressources sont les mêmes.

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

Ce troisième schéma d'axiomes contraint le signe des célérités d'une entité v lorsque ces célérités ont les mêmes ressources et contraint la variable v à tendre vers un état d'équilibre où la célérité applicable vaut 0.

Comme démontré dans Behaegel *et al.* (2017) et Behaegel (2018), cette logique de Hoare hybride est, elle aussi, correcte et complète. En d'autres mots, la précondition calculée par le calcul proposé est bien la plus faible précondition.

1.7.5. Holmes BioNet: une implémentation de la chaîne de traitements

Holmes BioNet (HOare Logic Modelling ExperimentS on Biological Networks) est un prototype qui construit automatiquement des contraintes sur les paramètres dynamiques d'un réseau de régulation hybride à partir d'une trace observée exprimée sous la forme d'un triplet de Hoare hybride. Il construit un problème de satisfaction de contraintes ou CSP grâce au calcul de plus faible précondition. Il utilise ensuite un solveur de contraintes sur les flottants pour identifier un ou plusieurs jeux de paramètres satisfaisant ces contraintes. Enfin il simule l'évolution du modèle dans le but de comparer avec les traces expérimentales, Cf. Behaegel *et al.* (2018b,a). Ces différentes étapes sont schématisées en figure 1.17. Holmes BioNet vise ainsi à l'automatisation de l'identification des paramètres et à la minimisation de l'intervention du modélisateur dans la construction du modèle. Les données d'entrée sont d'une part le réseau de régulation hybride, et d'autre part soit un triplet de Hoare ou a minima une postcondition et une trace expérimentale exprimée dans le langage de chemins.

La première étape consiste à utiliser le calcul de la plus faible précondition à partir de la post-condition donnée en remontant le chemin décrivant la trace biologique observée. Pour les modèles à comportement cyclique, sont ajoutées les contraintes spécifiant que l'état hybride initial du chemin doit être identique à l'état hybride final. Les contraintes sont simplifiées et réduites à la volée.

L'étape suivante fait appel au solveur de contraintes AbSolute de Pelleau *et al.* (2013) qui aide à l'identification d'un jeu de paramètres satisfaisant les contraintes. Toute solution donnée par AbSolute est représentée par deux listes: la première contient le domaine des variables dites sûres (quelque soit la valeur prise dans ce domaine, il existe une solution), et la seconde liste contient les domaines pouvant éventuellement contenir une solution. Par abus de langage, on parlera de liste d'invariants (variables sûres) et de liste de changeants (variables non sûres). Si la seconde liste est vide (s'il n'y a que des variables sûres), la solution est dite sûre et un jeu de paramètres peut simplement être choisi, variable par variable. Dans la plupart des cas, plusieurs solutions sont exhibées, elles peuvent être sûres ou non.

L'utilisateur spécifie un nombre souhaité de solutions et le solveur en exhibe autant lorsque c'est possible. S'il n'y a pas de solutions sûres, Holmes BioNet choisit une solution non sûre au hasard, une variable parmi les changeants est sélectionnée aléatoirement, on lui affecte une valeur arbitraire dans son domaine de définition, et cette variable devient constante (dans la liste des invariants). Cette contrainte est ensuite ajoutée au CSP, et le solveur est de nouveau appelé pour calculer les nouvelles solutions de ce nouveau système de contraintes. Ce procédé est répété jusqu'à épuiser la liste des changeants. Lorsqu'une solution sûre existe et que l'un de ses invariants présente un domaine de définition dont les bornes sont distinctes, sa valeur est choisie aléatoirement dans son intervalle de définition.

Enfin, la dernière étape consiste à réaliser une simulation de la dynamique du réseau de gènes en suivant les principes de la section 1.6.2.

1.8. Méthodologie générale

En conclusion, les approches développées dans ce chapitre s'intègrent dans une méthode générale de modélisation des réseaux de régulation biologique. Cette méthode, qui pourrait sans doute s'étendre à presque toutes les sciences expérimentales, se fonde sur une vision poppérienne, voir Popper (1979), de la biologie expérimentale:

- Lorsqu'un biologiste mène une campagne expérimentale, c'est pour tester une *hypothèse* scientifique qu'il considère importante vis à vis de la dynamique de son système.
- La modélisation lui sert de révélateur des différentes causalités et a pour objet, selon nous, de l'aider à concevoir ses expériences en vue de tester cette hypothèse.

– La fameuse *prédictivité* des modèles, qui est souvent considérée comme la principale qualité requise d'un modèle, n'est donc pas seulement la capacité à prévoir les trajectoires à partir d'états initiaux bien choisis, mais plus généralement la capacité à *suggérer des expériences* aptes à *réfuter* l'hypothèse si elle est fausse.

– Plusieurs modèles et *a fortiori* plusieurs trajectoires sont généralement compatibles avec l'hypothèse du biologiste et ce que l'on doit tenter de réfuter par des expériences, ce sont plutôt les *propriétés observables* conséquences de l'hypothèse, et cette vision est plus générale qu'une « simple » comparaison des résultats expérimentaux avec une trajectoire prédite (et la seule viable dans un contexte non déterministe).

La modélisation commence par exploiter les *connaissances* biologiques (hors hypothèse donc). Ces connaissances, généralement issues de la littérature, peuvent être de nature

– *structurelle*: cela permet de déterminer les variables pertinentes pour la question biologique abordée ainsi que les influences élémentaires ou combinées, conduisant à construire le graphe d'influences,

– *dynamique*: ce sont par exemple des homéostasies connues, des bassins d'attraction observés, des propriétés d'atteignabilité, ou plus généralement des comportements établis, que l'on peut formaliser en CTL',

– enfin *expérimentale*: ce sont typiquement des traces observées au cours d'une expérience, pouvant alors être traduites par un triplet de Hoare.

L'étape suivante consiste à vérifier la cohérence des connaissances ainsi formalisées: *TotemBioNet* permet d'automatiser cette étape. En pratique, si aucune paramétrisation n'est trouvée, cette incohérence peut être due soit à une extrapolation abusive d'un résultat extrait de la littérature, soit à une traduction abusive en CTL' ou en triplet de Hoare, soit encore à un graphe d'influences incomplet (ce dernier cas étant le plus souvent une réfutation très fructueuse pour la compréhension du système étudié). Ces différentes mises au point effectuées, *TotemBioNet* fournit un ensemble de paramétrisations possibles du réseau de régulation. Cet ensemble est non vide et généralement d'assez grande taille.

C'est maintenant le moment d'ajouter l'hypothèse, la plupart du temps exprimée en CTL'. Il suffit alors de réitérer l'identification des paramètres par *TotemBioNet* avec l'union des connaissances et de l'hypothèse.

– Si l'ensemble de paramétrisations obtenu est *vide* après s'être assuré de la bonne formalisation de l'hypothèse, on a réfuté l'hypothèse. Cela conduit le biologiste à revoir ses convictions, et à mieux préciser sous quelles conditions son hypothèse devrait s'appliquer. Là encore il s'agit en pratique d'une réfutation très fructueuse.

– Si l'ensemble de paramétrisations obtenu est *non vide*, la pratique nous montre que le nombre de paramétrisations compatibles est généralement très grand, et c'est

souvent dû à une formalisation incomplète des connaissances. On extrait alors au hasard un modèle et une trajectoire au sein de ce modèle et on la soumet aux experts biologistes. La réaction, la plupart du temps est radicale: cette trajectoire est impossible pour telle ou telle raison! Il suffit alors d'encoder ces raisons en CTL' et de relancer la démarche.

A cette étape, il faut bien garder en tête que la *cohérence* de l'hypothèse ne signifie en aucun cas sa *vérité*. L'ensemble des paramétrisations compatibles avec l'hypothèse est non vide mais la paramétrisation qui représente fidèlement le système biologique étudié n'a aucune raison d'appartenir à ce dernier: on sait seulement qu'elle appartient à l'ensemble des paramétrisations compatibles avec les connaissances.

C'est donc maintenant qu'il faut proposer des expériences. Elles doivent être aptes à réfuter au mieux l'appartenance de la « véritable » paramétrisation (inconnue) à l'ensemble de celles vérifiant l'hypothèse.

Nous qualifierons d'*observable* toute formule CTL' pour laquelle une expérience peut déterminer *sans ambiguïté* si elle est vraie ou fausse pour le système biologique. Une discussion avec les expérimentateurs permet généralement de définir des motifs syntaxiques caractérisant l'ensemble des formules observables, formalisant au mieux les capacités expérimentales du laboratoire de biologie impliqué.

Suivant Popper, il faut maintenant que, si l'hypothèse est fausse, il existe au moins une expérience qui contredise l'hypothèse. Il s'agit donc de prouver le théorème suivant « l'ensemble des conséquences observables de l'hypothèse, en utilisant également les connaissances, implique l'hypothèse ». En l'état actuel des choses, on effectue cette preuve à la main, même si certains lemmes peuvent être délégués à *TotemBioNet*. En pratique, si l'on n'arrive pas à prouver ce théorème de manière suffisamment simple, c'est probablement que les capacités expérimentales sont insuffisantes pour l'hypothèse considérée et l'on cherche à les renforcer.

La dernière étape de la méthode consiste à proposer en premier lieu les conséquences observables de l'hypothèse (donc les expériences) qui ont le plus de chance d'être contredites si l'hypothèse est fausse. Trouver une bonne stratégie de suggestion d'expériences est un sujet de recherche à part entière. A l'heure actuelle, de manière empirique, on se focalise sur les paramètres dont l'interprétation n'est pas imposée par les connaissances, et l'on propose des expériences qui mettent en jeu des trajectoires qui dépendent de ces paramètres. Notez que ces expériences, une fois menées, fournissent souvent des triplets de Hoare qui, à leur tour, permettent de réduire l'espace de recherche...

Historiquement, la méthode décrite ici a précédé les formalisations de la théorie de Thomas présentée dans ce chapitre. Les *méthodes formelles* sont fondamentalement la

clef pour construire des modèles prédictifs, et nous sommes persuadés que l'approche est applicable pour accompagner la plupart des sciences expérimentales.

1.9. Remerciement

Ce chapitre résulte d'un long travail, fruit de nombreuses collaborations. Beaucoup de collègues ont participé à développer une méthodologie de modélisation réellement utile pour la biologie. René Thomas et Houssine Snoussi sont bien sûr les premiers que nous voudrions remercier. Leurs idées ont été au cœur de notre travail. Janine Guespin-Michel a été la première biologiste à prendre énormément de temps pour nous familiariser avec les fondements du raisonnement en biologie, suivie de nombreux autres biologistes que nous ne pouvons pas tous citer. Adrien Richard a été le premier doctorant puis collègue à nous avoir accompagné dans notre démarche de développer des méthodes formelles pour la modélisation en biologie. Plus récemment (depuis dix ans tout de même) Olivier Roux et Morgan Magnin sont ceux qui nous ont ouvert aux approches hybrides. Émilien Cornillon a fortement participé à la définition actuelle du formalisme hybride. Maxime Folchette a développé le premier prototype pour le calcul de plus faible précondition, et Jonathan Behaegel s'est attelé à développer la logique de Hoare hybride. Qu'ils en soient tous remerciés.

Les auteurs sont aussi reconnaissants à l'ANR, à Génopole@-Évry ainsi qu'à l'Idex *UCA^{Jedi}* de l'Université Côte d'Azur pour leur aide au travers de nombreux projets au fil du temps.

1.10. Bibliographie

- Behaegel, J. (2018), Modèles hybrides de réseaux de régulation: étude du couplage des cycles cellulaire et circadien, PhD thesis, Université Côte d'Azur.
- Behaegel, J., Comet, J.-P., Bernot, G., Cornillon, E., Delaunay, F. (2015), A hybrid model of cell cycle in mammals, *in* 6th International Conference on Computational Systems-Biology and Bioinformatics (CSBio'2015), Bangkok (Thailand).
- Behaegel, J., Comet, J.-P., Folschette, F. (2017), Constraint identification using modified Hoare logic on hybrid models of gene networks, *in* Proceedings of the 24th International Symposium on Temporal Representation and Reasoning (TIME), pp. 5:1–5:21.
- Behaegel, J., Comet, J.-P., Pelleau, M. (2018a), Identification de paramètres dynamiques de réseaux de gènes, *in* Quatorzièmes Journées Francophones de Programmation par Contraintes.
- Behaegel, J., Comet, J.-P., Pelleau, M. (2018b), Identification of dynamic parameters for gene networks, *in* Proceedings of the 30th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Volos, Greece.

- Bernot, G., Comet, J.-P., Khalis, Z., Richard, A., Roux, O. F. (2018), A genetically modified Hoare logic, *Theoretical Computer Science*, . <https://doi.org/10.1016/j.tcs.2018.02.003>.
- Bernot, G., Comet, J.-P., Richard, A., Guespin, J. (2004), Application of formal methods to biological regulatory networks: Extending Thomas' asynchronous logical approach with temporal logic, *Journal of Theoretical Biology*, 229(3), 339–347.
- Cimatti, A., Clarke, E., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., Sebastiani, R., Tacchella, A. (2002), NuSMV 2: An opensource tool for symbolic model checking, in *Proceedings of the 14th International Conference on Computer Aided Verification, CAV '02*, Springer-Verlag, London, UK, UK, pp. 359–364.
- Clarke, E., Emerson, E. (1981), Design and syntheses of synchronization skeletons using branching time temporal logic, in *Proc. Logics of Programs Workshop*, Yorktown Heights, New York, vol. 131 of *LNCS*, Springer, pp. 52–71.
- Dijkstra, E. (1975), Guarded commands, nondeterminacy and formal derivation of programs, *Commun. ACM*, 18, 453–457.
- D'Silva, V., Kroening, D., Weissenbacher, G. (2008), A survey of automated techniques for formal software verification, *IEEE Trans. on CAD of Integrated Circuits and Systems*, 27(7), 1165–1178.
- Folschette, M. (2011), Application de la logique de Hoare aux réseaux de régulation génétique avec multiplexes, Master's thesis, Institut de Recherche en Communications et Cybernétique de Nantes.
- Hoare, C. (1969), An axiomatic basis for computer programming, *Communications of the ACM*, 12(10), 576–585.
- Huth, M., Ryan, M. (2000), *Logic in Computer Science: Modelling and reasoning about systems*, Cambridge University Press.
- Khalis, Z. (2010), Logique de Hoare et identification formelle des paramètres d'un réseau génétique, PhD thesis, Université d'Evry-Val d'Essonne.
- Khalis, Z., Bernot, G., Comet, J.-P. (2009), *Proc. of the Nice Spring school on Modelling complex biological systems in the context of genomics*, EDP Science, ISBN : 978-2-7598-0437-5, chapter Gene Regulatory Networks: Introduction of multiplexes into R. Thomas' modelling, pp. 139–151.
- Khalis, Z., Comet, J.-P., Richard, A., Bernot, G. (2009), The SMBioNet method for discovering models of gene regulatory networks, *Genes, Genomes and Genomics*, 3(special issue 1), 15–22.
- Khoodeeram, R., Bernot, G., Trosset, J.-Y. (2017), *An Ockham Razor model of energy metabolism*, EDP Science publisher, ISBN: 978-2-7598-2116-7, chapter Book chapter in *Proc. of the Thematic Research School on Advances in Systems and Synthetic Biology*, pp. 81–101.
- McMillan, K. (1993), *Symbolic Model Checking*, Kluwer Academic Publishers.

- Pelleau, M., Miné, A., Truchet, C., Benhamou, F. (2013), A constraint solver based on abstract domains, *in* R. Giacobazzi, J. Berdine, I. Mastroeni, (eds), *Verification, Model Checking, and Abstract Interpretation*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 434–454.
- Popper, K. (1979), *Conjectures et réfutations, La croissance du savoir scientifique*, Payot.
- Pérès, S., Comet, J.-P. (2003), Contribution of computational tree logic to biological regulatory networks: example from *pseudomonas aeruginosa*, *in* International workshop on Computational Methods in Systems Biology, vol. 2602 of *LNCIS*, pp. 47–56.
- Richard, A. (2006), *Modèle formel pour les réseaux de régulation génétique et influence des circuits de rétroaction*, PhD thesis, Université d'Evry-Val d'Essonne.
- Richard, A. (2008), Fair paths in CTL. Note interne I3S.
- Richard, A., Comet, J.-P., Bernot, G. (2006), *Modern Formal Methods and Applications*, Springer, ISBN: 1-4020-4222-1, chapter Formal Methods for Modeling Biological Regulatory Networks, pp. 83–122.
- Snoussi, E. H. (1989), Qualitative dynamics of piecewise-linear differential equations: a discrete mapping approach, *Dynamics and stability of Systems*, 4(3-4), 565–583.
- Snoussi, E., Thomas, R. (1993), Logical identification of all steady states : the concept of feedback loop characteristic states, *Bull. Math. Biol.*, 55(5), 973–991.
- Thomas, R. (1978), Logical analysis of systems comprising feedback loops., *J. Theor. Biol.*, 73(4), 631–56.
- Thomas, R. (1991a), Regulatory networks seen as asynchronous automata : A logical description, *J. theor. Biol.*, 153, 1–23.
- Thomas, R. (1991b), Regulatory networks seen as asynchronous automata: a logical description, *Journal of theoretical biology*, 153(1), 1–23.
- Thomas, R., d'Ari, R. (1990), *Biological Feedback*, CRC Press.
- Zhang, R., Lahens, N., Ballance, H., Hughes, M., Hogenesch, J. (2014), A circadian gene expression atlas in mammals: implications for biology and medicine., *PNAS*, 111(45), 16219–24.