

# Identification de paramètres dynamiques de réseaux de gènes

Jonathan Behaegel<sup>1\*</sup> Jean-Paul Comet<sup>1</sup> Marie Pelleau<sup>1</sup>

<sup>1</sup> Université Côte d'Azur, CNRS, I3S, France

behaegel@i3s.unice.fr comet@unice.fr marie.pelleau@i3s.unice.fr

## Résumé

L'étude des réseaux de gènes nous permet de mieux comprendre certains processus biologiques tels que l'adaptation de l'organisme à une perturbation de l'environnement. Dans le cadre d'une modélisation discrète des réseaux de gènes, il a été montré que la logique de Hoare pouvait aider à l'identification des paramètres du modèle afin que ce dernier exhibe les traces biologiques observées. Dans cet article nous présentons une modélisation hybride des réseaux de gènes mettant l'accent sur le temps passé dans chacun des états et nous introduisons une extension de la logique de Hoare dans ce cas hybride. Le calcul de la plus faible précondition associée à cette logique de Hoare modifiée permet de déterminer les contraintes minimales sur les paramètres dynamiques d'un réseau de gènes à partir d'une trace biologique observée. Ces contraintes forment un CSP continu que nous résolvons à l'aide du solveur continu AbSolute. Les premiers résultats expérimentaux montrent que les solutions exhibées sont en accord avec les spécifications du triplet de Hoare provenant de l'expertise biologique.

## Abstract

The study of gene networks allows us to better understand some biological processes such as the adaptation of the organism to a disturbance of the environment. In a discrete modelling framework of gene networks, it has been shown that the Hoare logic can help the modeller to identify the parameters of the model so that the latter exhibits the observed biological traces. In this paper we present a hybrid modelling of gene networks which pays particular attention to the time spent in each state and we introduce an extension of the Hoare logic in this hybrid case. The weakest precondition calculus associated with this modified Hoare logic makes it possible to determine the minimal constraints on the dynamic parameters of a gene network from an observed biological trace. These constraints form a continuous CSP that we

solve using the AbSolute continuous solver. The first experimental results show that the obtained solutions are in agreement with the specification of the Hoare triple coming from biological expertise.

## 1 Introduction

La modélisation des réseaux de régulation génétique a pour but l'étude et la compréhension des mécanismes moléculaires permettant à l'organisme d'assurer les fonctions essentielles allant du métabolisme à l'adaptation à une perturbation de l'environnement. Les réseaux de gènes sont décrits par des régulations qui peuvent être classées en deux types : les activations et les inhibitions. La combinaison de ces régulations permettent de nombreux comportements et on peut montrer que la complexité de ces systèmes provient des rétrocontrôles positifs et négatifs couramment observés qui mènent à la multistationnarité et à l'homéostasie (capacité à maintenir un équilibre). L'étude de la dynamique de ces systèmes ouvre de nouvelles perspectives avec des applications en pharmacologie, en médecine, ou en toxicologie... Différents cadres de modélisation (qualitative, continue, stochastique, hybride) sont possibles mais quel que soit le cadre choisi, un point crucial du processus de modélisation réside dans la détermination des paramètres qui gouvernent la dynamique du modèle et leur identification reste l'étape limitante.

Le cadre de modélisation discret de René Thomas [17] ne cherche à représenter que l'enchaînement d'événements qualitatifs et les méthodes formelles classiques (model checking, programmation par contraintes, logique de Hoare) ont été utilisées avec succès pour automatiser l'identification des paramètres gouvernant la dynamique [4, 7, 3]. Cependant, certains phénomènes biologiques font intervenir une

\*Papier doctorant : Jonathan Behaegel<sup>1</sup> est auteur principal.

composante temporelle jouant un rôle primordial : le cycle circadien en est un bel exemple puisqu'il permet à l'organisme de s'adapter à l'alternance jour/nuit. Pour modéliser ces phénomènes, la prise en compte du temps impose *a minima* un cadre de modélisation hybride qui ajoute à l'approche de René Thomas une mesure du temps passé dans chacun des états. Notre objectif est alors de chercher à automatiser l'identification des paramètres d'un modèle hybride pour que ce dernier soit en accord avec les observations expérimentales (variations de la concentration des protéines d'un système biologique au cours de la journée, données très difficiles à obtenir pour un grand nombre de protéines). Ces observations sont représentées par une trace biologique qui exprime un enchaînement d'événements qualitatifs séparés par les durées mesurées. Pour cela, nous avons modifié la logique de Hoare ainsi que son calcul de plus faible précondition pour construire un système de contraintes résolue par le solveur AbSolute [14].

La programmation par contraintes a déjà été utilisée pour résoudre des problèmes liés aux réseaux biologiques. Dans le cadre discret, elle a été utilisée pour la recherche des paramètres cinétiques [7], ou la recherche de bifurcations [10]. Elle a aussi été utilisée pour l'optimisation de modes de flux dans des réseaux métaboliques [13]. Dans le cadre continu, l'approche par contraintes a permis de vérifier des propriétés temporelles du modèle [5], ou identifier les paramètres dynamiques d'un système décrit par équations différentielles dans le cadre de la biologie synthétique [15]. Notre but est d'aborder pour des modèles hybrides le problème d'identification des paramètres dynamiques alors que généralement ce sont les problèmes de vérification [1, 12] ou d'approximation de modèle [11] qui sont abordés.

L'article présente en section 2 le cadre de modélisation hybride utilisé pour décrire la dynamique des réseaux de gènes. La section 3 décrit la logique de Hoare ainsi que son calcul de plus faible précondition adapté à ce cadre de modélisation hybride. Le solveur AbSolute est esquissé en section 4 en mettant l'accent sur la réécriture symbolique et la notion de solution sûre. Nous présentons en section 5 nos résultats expérimentaux d'identification de paramètres dynamiques grâce à AbSolute et nous finissons par discuter l'approche proposée.

## 2 Modélisation d'un réseau de gènes

De nombreux cadres de modélisation dédiés aux réseaux génétiques peuvent être utilisés pour représenter la dynamique du réseau de gènes. Un des premiers cadres de modélisation qui a été totalement forma-

lisé est celui introduit par René Thomas [17, 4] mais le temps séparant deux événements qualitatifs a été totalement abstrait. Ici, nous présentons un cadre de modélisation hybride inspiré de l'approche discrète de Thomas [2].

### 2.1 Représentation d'un réseau de gènes

Les réseaux de gènes sont représentés par des graphes étiquetés, voir figure 1. Les deux types de sommets permettent de décrire d'une part les entités (abstractions de gènes et protéines associés, représentées par des cercles) et d'autre part les régulations entre ces entités (représentées par des rectangles) et appelées multiplexes. L'ensemble des entités est noté  $V$ . Les arcs pointant sur un multiplexe (arcs en pointillés) spécifient les entités prenant part à la régulation et l'ensemble des prédécesseurs d'une entité  $v \in V$  est noté  $R^-(v)$ . Les arcs sortant d'un multiplexe (arcs pleins) spécifient les entités qui sont *positivement* régulées par le multiplexe. Comme les régulations ne sont pas toujours actives, on associe à chaque multiplexe une formule décrivant les conditions sous lesquelles la régulation associée au multiplexe a lieu.

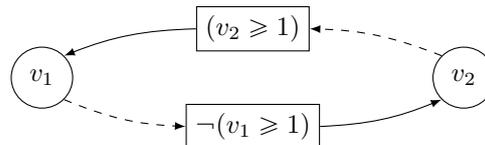


FIGURE 1 – Graphe d'interaction représentant une boucle de rétroaction négative simple.

La figure 1 présente un graphe d'interaction avec 2 entités  $v_1$  et  $v_2$  :  $v_2$  active  $v_1$  et  $v_1$  inhibe  $v_2$  (l'inhibition est représentée par la négation de la formule). Le tout forme une boucle de rétroaction négative puisque chacune des 2 entités a une action négative indirecte sur elle-même.

### 2.2 Dynamique d'un réseau de gènes

Le cadre de modélisation hybride [2] repose sur la discrétisation des concentrations des entités biologiques, tout en incluant le temps passé entre deux événements successifs. Supposons que l'entité  $v$  agisse sur d'autres entités à différents seuils et que le nombre de seuils différents soit  $b_v$ . Alors l'ordonnancement de ces seuils mène à la définition des niveaux qualitatifs de l'entité  $v$  : le niveau 0 représente l'intervalle des concentrations inférieures au premier seuil, le niveau  $b_v$  représente l'intervalle des concentrations supérieures au plus grand seuil et le niveau  $k$  représente l'intervalle des concentrations comprises entre le  $k$ -ème et le

$(k+1)$ -ème seuil ( $k \in \llbracket 0, b_v \rrbracket$ ). Un état qualitatif du système est la donnée d'un niveau qualitatif pour chacune des entités.

A chaque état qualitatif est associé un "espace temporel" qui peut être vu comme l'hypercube  $[0, 1]^n$  où  $n$  est le nombre d'entités, voir les carrés de la figure 2. L'état du système est caractérisé non seulement par l'état qualitatif, mais aussi par la position précise au sein de l'espace temporel. Les états du système sont ainsi appelés états hybrides et sont définis par un couple de vecteurs  $h = (\eta, \pi)$  où  $\eta$ , appelé état discret, est le vecteur des niveaux qualitatifs des entités (la composante  $v$  du vecteur est un entier positif inférieur ou égal à  $b_v$ ) et  $\pi \in [0, 1]^n$ , appelé partie fractionnaire, est le vecteur des positions exactes à l'intérieur de l'espace temporel associé à cet état discret, voir figure 2. Par convention,  $\eta_v$  (resp.  $\pi_v$ ) dénote le niveau qualitatif (resp. la partie fractionnaire) de l'entité  $v$ . Deux états discrets seront dits voisins s'ils ne diffèrent que d'une seule composante et si cette composante ne diffère que de un.

L'évolution au sein de chaque état qualitatif, représentée par des *transitions continues*, est gouvernée par des célérités à identifier, et les transitions d'un état à un autre appelées *transitions discrètes* sont possibles sous certaines conditions.

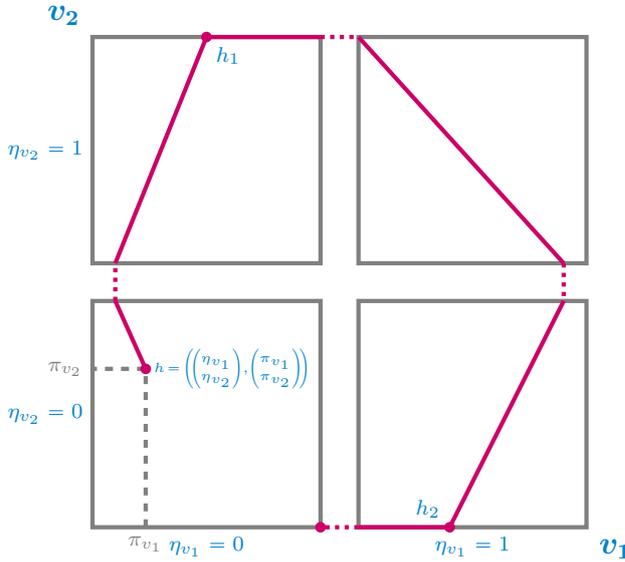


FIGURE 2 – Une dynamique possible associée au graphe d'interaction de la figure 1.

La figure 2 montre l'évolution d'un modèle possible pour le graphe d'interaction de la figure 1 en partant du point  $h$  à l'intérieur de l'état discret  $(\eta_{v_1}, \eta_{v_2}) = (0, 0)$ . Tant qu'on reste dans cet état discret, la concentration de  $v_2$  augmente. Lorsque  $v_2$  ne peut plus augmenter, on change d'état discret  $(\eta_{v_1}, \eta_{v_2}) = (0, 1)$ .

Dans ce nouvel état,  $v_1$  et  $v_2$  augmentent jusqu'à saturation de  $v_2$ .  $v_1$  continue alors d'augmenter ce qui permet d'entrer dans l'état discret  $(1, 1)$ ... Finalement la trace de la figure 2 s'arrête dans l'état discret  $(0, 0)$  où  $\pi_{v_1} = 1$  et  $\pi_{v_2} = 0$ .

Un multiplexe est actif dans un état hybride  $h$  si sa formule associée est évaluée à vrai après substitution des entités par leurs niveaux qualitatifs. Les multiplexes actifs seront appelés *ressources* et l'ensemble des ressources de  $v$  à l'état  $h$  est noté  $\rho(h, v)$ . Par exemple, dans la figure 2 on a  $\rho(h, v_1) = \emptyset$  car la formule du seul multiplexe régulant  $v_1$  est évaluée à faux dans cet état, et  $\rho(h, v_2) = \{v_1\}$  car la formule du seul multiplexe régulant  $v_2$  est évaluée à vrai dans cet état. Notons que les ressources d'une entité à l'état hybride  $h$  ne dépendent que des niveaux qualitatifs des entités : les ressources ne varient pas à l'intérieur d'un "espace temporel".

Les différentes entités n'évoluent évidemment pas à la même vitesse dans chacun des "espaces temporels". De même, la vitesse d'une entité dépend de ses ressources et de son niveau qualitatif. Ainsi, la dynamique des entités est contrôlée par les célérités  $C_{v,\omega,k} \in \mathbb{R}$  indexées par l'entité  $v$ , par  $\omega$  l'ensemble des ressources de  $v$  dans l'état discret courant, et par le niveau qualitatif  $k$  de l'entité  $v$ . Par exemple, dans l'état discret  $(0, 0)$  de la figure 2, la célérité contrôlant l'entité  $v_1$  est  $C_{v_1, \emptyset, 0}$  et la célérité contrôlant l'entité  $v_2$  est  $C_{v_2, \{v_1\}, 0}$ . Une célérité nulle indique que l'entité a atteint un état d'équilibre. Pour atteindre cet état d'équilibre, il faut de plus que les célérités de cette entité, soumises aux mêmes régulations, se dirigent vers cet état d'équilibre :

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

Dans tous les cas, les célérités d'une même entité avec les mêmes ressources dans des états discrets voisins se dirigent dans la même direction et donc ont le même signe :

$$\forall v \in V, \forall \omega \subset R^-(v), \forall k \in \llbracket 0, b_v - 1 \rrbracket, \quad C_{v,\omega,k} \times C_{v,\omega,k+1} \geq 0 \quad (2)$$

Dans un état discret, si la célérité de  $v$  est positive (resp. négative), on dit que  $v$  fait face à un mur externe lorsque le niveau qualitatif maximal (resp. minimal) est atteint  $\eta_v = b_v$  (resp.  $\eta_v = 0$ ). Ainsi,  $v$  ne peut pas continuer à augmenter (resp. diminuer). Dans les autres cas, si  $\eta_v < b_v$  (resp.  $\eta_v > 0$ ), l'entité  $v$  fait face à un mur interne lorsque les célérités de  $v$  dans l'état courant et dans l'état voisin vers lequel  $v$  tendrait, sont de signes contraires.

Enfin, les transitions entre deux états discrets sont possibles lorsque les deux conditions suivantes sont respectées : une entité  $v$  doit être sur le bord de l'état discret ( $\pi_v = 0$  ou  $1$ ) et cette entité doit pouvoir changer de niveau qualitatif (ne pas faire face à un mur). La définition 1 introduit la notion d'entité glissante :

**Définition 1** Soit un réseau de gènes,  $v \in V$  une entité et  $h = (\eta, \pi)$  un état hybride.

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

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

$$((C_{v,\rho(h,v),\eta_v} > 0) \wedge (\eta_v = b_v)) .$$
2. Soit  $h' = (\eta', \pi')$  un autre état hybride tel que  $\eta'_v = \eta_v + \text{sgn}(C_{v,\rho(h,v),\eta_v})$  et  $\eta'_u = \eta_u$  pour tout  $u \in V, u \neq v$ .  $v$  fait face à un mur interne en  $h$  si  $\text{sgn}(C_{v,\rho(h,v),\eta_v}) \times \text{sgn}(C_{v,\rho(h',v),\eta'_v}) = -1$ , avec  $\text{sgn}$  la fonction signe usuelle,  $\rho(h, v)$  les ressources de  $v$  dans  $h$ .

On note  $\text{sv}(h)$  l'ensemble des entités glissantes, c.-à-d. qui font face à un mur interne ou externe en  $h$ .

Dans la figure 2, l'entité  $v_2$  fait face à un mur externe, elle est donc glissante dans les états discrets  $(0, 1)$  et  $(1, 0)$ .

Pour chaque entité, le temps pour atteindre un bord est appelé délai de contact, qui dépend de la célérité de l'entité et de la distance à parcourir dans l'état hybride  $h$  (définition 2). Notons que lorsque la célérité d'une entité est nulle, le délai de contact est infini.

**Définition 2 (Délai de contact)** Soit un réseau de gènes,  $v$  une entité de  $V$  et  $h = (\eta, \pi)$  un état hybride. Nous notons  $\delta_h(v)$  le délai de contact de  $v$  en  $h$  pour atteindre le bord de l'état discret.  $\delta_h$  est la fonction de  $V$  dans  $\mathbb{R}^+ \cup +\infty$  définie par :

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

Nous définissons également l'ensemble *first* des entités atteignant leur bord en premier à partir de l'état hybride  $h$  (c.-à-d. les entités ayant un délai de contact minimum dans l'état hybride courant), en excluant les entités *glissantes*. Ainsi, l'ensemble *first* représente l'ensemble des entités qui vont pouvoir changer de niveau qualitatif et mener à des transitions discrètes.

La définition complète de la dynamique d'un réseau de gènes (transitions continues et transitions discrètes) est donnée dans [2]. La partie suivante introduit un moyen de construire les contraintes sur les paramètres dynamiques d'un réseau de gènes.

### 3 Logique de Hoare

La logique de Hoare a été développée pour prouver la correction des programmes impératifs [9]. Elle s'appuie sur la notion de triplet de Hoare de la forme  $\{Pre\} p \{Post\}$  où  $Pre$  et  $Post$  représentent des conditions sur l'état du système et  $p$  est un programme impératif. D'un point de vue intuitif, un triplet sera dit correct si l'exécution du programme  $p$ , à partir d'un état où la précondition  $Pre$  est satisfaite, est possible et mène à un état où la postcondition  $Post$  est satisfaite. Les règles d'inférence associées à l'affectation, l'instruction conditionnelle, la composition de programme et le *while* permettent de prouver efficacement qu'un programme, s'il termine, est correct.

Dans cet article, nous modifions la logique de Hoare pour pouvoir synthétiser les contraintes nécessaires sur les paramètres dynamiques du modèle assurant que le modèle exhibe une trace biologique observée. Cette approche a été développée dans le cadre de modélisation discrète [3] et nous présentons ici l'extension au cadre hybride présentée dans [2].

#### 3.1 Logique de Hoare modifiée

Dans le cadre de la modélisation des réseaux de gènes, le programme impératif  $p$  est remplacé par une trace biologique observée écrite de manière formelle dans le langage de chemins, et les conditions  $Pre$  et  $Post$  expriment des propriétés sur les états hybrides de départ et d'arrivée.

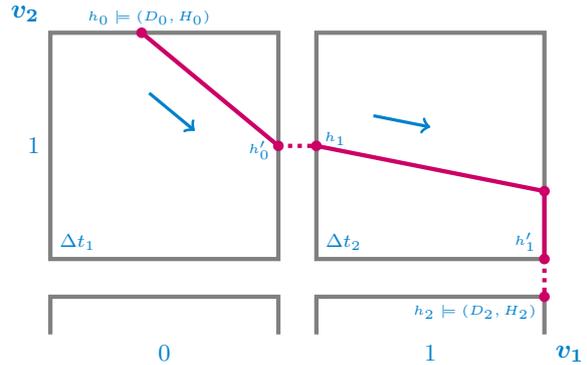


FIGURE 3 – Trace biologique avec des transitions continues et discrètes. Les pré et postconditions sont  $(D_0, H_0)$  et  $(D_2, H_2)$ .  $\Delta t_1$  et  $\Delta t_2$  représentent les temps passés dans les états qualitatifs et les vecteurs représentent les vecteurs célérités.

Les pré et postconditions sont exprimés dans un langage de propriété : une propriété est un couple de la forme  $(D, H)$ , où la condition discrète  $D$  décrit exclusivement l'état discret  $\eta$  de chaque entité biologique, et

la condition hybride  $H$  porte sur les parties fractionnaires  $\pi$  de ces mêmes entités. Si aucune connaissance sur la condition  $D$  (resp.  $H$ ) n'est disponible, cette dernière est représentée par la tautologie True. La figure 3 montre un chemin dont le point de départ  $h_0$  est défini par la précondition  $(\eta(v_1) = 0 \wedge \eta(v_2) = 1, \pi_{v_1} = 0.4 \wedge \pi_{v_2} = 1)$ .

Le langage de chemins permet de décrire les traces biologiques : il comporte le chemin vide  $\epsilon$ , les chemins élémentaires qui sont de la forme  $(\Delta t, a, dpa)$  et les chemins non élémentaires composés d'une séquence finie de chemins élémentaires. Un chemin élémentaire est défini par un triplet  $(\Delta t, a, dpa)$  :

- L'atome de chemin discret  $dpa$  (discrete path atom), de la forme  $v+$  (resp.  $v-$ ), exprime la prochaine transition vers un état discret voisin : le niveau qualitatif de  $v$  est incrémenté (resp. décrémenté).
- L'élément  $a$  permet de spécifier des propriétés relatives à la dynamique au sein de l'état discret courant. Le modélisateur peut caractériser les comportements à l'aide du langage d'assertion suivant :  $a ::= \top \mid C_v \sqcap c \mid (\text{no})\text{slide}(v) \mid (\text{no})\text{slide}^+(v) \mid (\text{no})\text{slide}^-(v) \mid \neg a \mid a \wedge a \mid a \vee a$  où  $\sqcap$  appartient à l'ensemble des symboles de comparaison. Le terme  $\text{slide}^+(v)$  (resp.  $\text{slide}^-(v)$ ) indique que  $v$  va glisser le long de son bord supérieur (resp. inférieur), tandis que  $\text{noslide}^+(v)$  (resp.  $\text{noslide}^-(v)$ ) empêche ce glissement. Le terme  $\text{slide}(v)$  (resp.  $\text{noslide}(v)$ ) désigne la disjonction (resp. conjonction) de  $\text{slide}^+(v)$  et  $\text{slide}^-(v)$  (resp.  $\text{noslide}^+(v)$  et  $\text{noslide}^-(v)$ ). Notons que dans un chemin élémentaire, une même entité ne peut apparaître à la fois dans le  $dpa$  et dans un terme  $\text{slide}$ . Pour exprimer que la trace de la figure 3 possède un glissement de  $v_1$  dans l'état discret où  $\eta_{v_1} = 1$  et  $\eta_{v_2} = 1$ , on peut écrire  $\text{slide}^+(v_1)$ . Ce langage d'assertion reprend le langage défini en [2] en le complétant par les atomes  $\text{noslide}^\pm(v)$  qui permettent de mieux décrire les données expérimentales.
- Enfin,  $\Delta t$  indique le temps exact passé dans l'état discret courant.

Ainsi, le triplet de Hoare représentant la trace de la figure 3 s'écrit :

$$\left\{ \begin{array}{l} D_0 \\ H_0 \end{array} \right\} \left( \begin{array}{c} \Delta t_1 \\ \top \\ v_1+ \end{array} \right); \left( \begin{array}{c} \Delta t_2 \\ \text{slide}^+(v_1) \\ v_2- \end{array} \right) \left\{ \begin{array}{l} D_2 \\ H_2 \end{array} \right\} \quad (3)$$

On dira qu'une transition continue  $h \rightarrow h'$  satisfait le couple d'assertion  $(\Delta t, a)$  si la transition existe, si cette transition dure  $\Delta t$  et si elle respecte l'assertion  $a$  ( $C_v \sqcap c$  est satisfait si  $C_{v,\rho(h,v),\eta_v} \sqcap c$ ,  $\text{slide}^+(v_1)$  est satisfait si  $\pi_{v_1}$  atteint 1 et  $v_1$  fait face à un mur, etc.)

Chaque instruction élémentaire du chemin  $p$  correspond à une transition continue d'une durée  $\Delta t$  définie dans l'état discret courant, suivi d'une transition discrète permettant à l'entité du  $dpa$  d'incrémenter ou de décrémenter son niveau qualitatif. La figure 3 montre qu'à partir de l'état hybride  $h_0$  satisfaisant la précondition de l'équation 3, il existe une transition continue  $h_0 \rightarrow h'_0$  d'une durée  $\Delta t_1$  menant à une transition discrète  $h'_0 \rightarrow h_1$ . Le processus est répété pour l'instruction élémentaire suivante où un glissement en  $v_1$  ( $\text{slide}^+(v_1)$ ) a lieu, jusqu'à atteindre l'état hybride  $h_2$  satisfaisant la postcondition  $(D_2, H_2)$ .

### 3.2 Plus faible précondition

Edsger Dijkstra a introduit une sémantique de transformation de prédicats [9] pour les langages de programmation impérative : elle associe à chaque instruction de ce langage une transformation de prédicats. Pour chaque instruction élémentaire, la plus faible précondition est une fonction qui associe à chaque postcondition  $Post$  une précondition  $Pre$ . En itérant le processus, on peut construire la plus faible précondition d'un programme impératif complet. Nous définissons de manière similaire la plus faible précondition pour chacun de nos chemins élémentaires :

**Définition 3 (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  $p = \epsilon$ , alors  $D_{i,f} \equiv D_f$  and  $H_{i,f} \equiv H_f$  ;
- Si  $p = (\Delta t, a, v^\pm)$ , alors :

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

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

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

indexée par un nouvel indice intermédiaire  $m$  ; avec  $\Phi_v^\pm(\Delta t)$ ,  $\mathcal{W}_v^\pm$ ,  $\mathcal{F}(\Delta t)$ ,  $\mathcal{A}(\Delta t, a)$  and  $\mathcal{J}_v$  des sous-propriétés définies en annexe 7.1.

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  $\Delta t$  et assurant la transition discrète;
- $\mathcal{F}(\Delta t)$  contraint toutes les entités sauf l'entité  $v$  du  $dpa$  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.

Ces 5 sous-propriétés sont décrites avec précision en annexe 7.1.

Sur la figure 3, dans l'état discret (1, 1),  $v_1$  atteint en premier un mur et on a un glissement jusqu'à ce que  $v_2$  atteigne son bord inférieur. La plus faible précondition associée au chemin  $p$  et à la postcondition  $post$  :

$$p = \begin{pmatrix} \Delta t_2 \\ a \\ v_2- \end{pmatrix} \quad post = \begin{Bmatrix} D_2 \\ H_2 \end{Bmatrix} \quad \text{avec } a \equiv \text{slide}^+(v_1)$$

est donnée par

$$\begin{aligned} D_1 &\equiv D_2[\eta_{v_2} \setminus \eta_{v_2} - 1], \\ H_1 &\equiv H_2[\eta_{v_2} \setminus \eta_{v_2} - 1] \wedge \Phi_{v_2}^-(\Delta t_2) \wedge \mathcal{F}(\Delta t_2) \wedge \\ &\quad \neg\mathcal{W}_{v_2}^- \wedge \mathcal{A}(\Delta t_2, a) \wedge \mathcal{J}_{v_2}; \end{aligned}$$

où les sous-propriétés font intervenir les parties fractionnaires de l'état d'entrée  $h_1$  ( $\pi_{v_1}^1$  et  $\pi_{v_2}^1$ ) ainsi que les parties fractionnaires de l'état de sortie  $h'_1$  ( $\pi_{v_1}^{1'}$  et  $\pi_{v_2}^{1'}$ ), et s'expriment après simplification comme suit :

- $v_2$  atteint son bord inférieur  
 $\Phi_{v_2}^-(\Delta t_2) \equiv (\pi_{v_2}^{1'} = 0) \wedge (C_{v_2, \rho(h'_1, v_2), 1} < 0) \wedge$   
 $(\pi_{v_2}^1 = \pi_{v_2}^{1'} - C_{v_2, \rho(h'_1, v_2), 1} \times \Delta t_2)$
- $v_2$  n'atteint pas de mur  
 $\neg\mathcal{W}_{v_2}^- \equiv C_{v_2, \rho(h_2, v_2), 0} \leq 0$
- $v_1$  atteint un mur grâce à  $\text{slide}^+(v_1)$   
 $\mathcal{F}(\Delta t_2) \equiv C_{v_1, \rho(h'_1, v_1), 1} > 0$   
 $\mathcal{A}(\Delta t_2, a) \equiv (\pi_{v_1}^1 > \pi_{v_1}^{1'} - C_{v_1, \rho(h'_1, v_1), 1} \times \Delta t_2)$   
 $\wedge (\pi_{v_1}^{1'} = 1)$
- Jonction entre les états  $h_2$  et  $h'_1$   
 $\mathcal{J}_{v_2} \equiv (\pi_{v_2}^2 = 1 - \pi_{v_2}^{1'}) \wedge (\pi_{v_1}^2 = \pi_{v_1}^{1'}) \wedge$   
 $(C_{v_1, \rho(h'_1, v_1), 1} > 0) \wedge$   
 $(\pi_{v_1}^1 \geq \pi_{v_1}^{1'} - C_{v_1, \rho(h'_1, v_1), 1} \times \Delta t_2)$

L'ensemble de ces sous-propriétés permet de décrire les contraintes sur les célérités et éventuellement comparer les célérités de différentes entités entre elles. Par exemple, les assertions  $\text{slide}^\pm(v_1)$  et  $\text{noslide}^\pm(v_1)$  apportent des contraintes entre la célérité de l'entité  $v_1$  et la célérité de l'entité  $v_2$  du  $dpa$  car elles sont toutes les deux reliées au temps passé dans l'état discret courant.

La stratégie choisie pour le calcul de la plus faible précondition est la même que celle proposée par Dijkstra [9], en partant de la postcondition et en remontant les chemins élémentaires jusqu'au premier (Définition 3). Une implémentation de cette stratégie a été réalisée<sup>1</sup> et la preuve de correction de la logique de Hoare adaptée au cadre de modélisation hybride est donnée dans [2].

### 3.3 Exemple de contraintes obtenues

En se basant sur la figure 1, on peut construire les contraintes sur les célérités de la boucle négative à partir d'une trace biologique expérimentale. Pour cela, on considère le chemin et la postcondition suivants :

$$\left( \begin{matrix} 5.0 \\ \text{nsI}(v_1) \\ v_2+ \end{matrix} \right); \left( \begin{matrix} 7.0 \\ \text{slide}^+(v_2) \\ v_1+ \end{matrix} \right); \left( \begin{matrix} 8.0 \\ \text{nsI}(v_1) \\ v_2- \end{matrix} \right); \left( \begin{matrix} 4.0 \\ \text{slide}^-(v_2) \\ v_1- \end{matrix} \right) \begin{Bmatrix} D_4 \\ H_4 \end{Bmatrix}$$

avec  $D_4 \equiv (\eta_{v_1} = 0) \wedge (\eta_{v_2} = 0)$  et  $H_4 \equiv \text{True}$  et  $\text{nsI}$  une abréviation du symbole  $\text{noslide}$ .

Nous appliquons la stratégie choisie du calcul de plus faible précondition sur le dernier chemin élémentaire

$$ce = \left( \begin{matrix} 4.0 \\ \text{slide}^-(v_2) \\ v_1- \end{matrix} \right) \text{ en utilisant la postcondition } \begin{Bmatrix} D_4 \\ H_4 \end{Bmatrix} \text{ et}$$

nous calculons  $\text{WP}(ce, (D_4, H_4)) \equiv (D_3, H_3)$ .

Le temps passé dans l'état courant est de 4.0 heures. Le  $dpa$   $v_1-$  indique que l'entité  $v_1$  atteint son bord inférieur et le franchit. L'assertion  $\text{slide}^-(v_2)$  quant à elle indique que l'entité  $v_2$  atteint son bord inférieur avant que  $v_1$  atteigne le sien.

D'après la définition 3, on peut facilement calculer les niveaux qualitatifs de chaque entité de la condition discrète à partir des niveaux qualitatifs observés en postcondition :

$$D_3 \equiv D_4[\eta_{v_1} \setminus \eta_{v_1} - 1] \equiv \eta_{v_1} = 1 \wedge \eta_{v_2} = 0$$

Les célérités à utiliser dans l'état courant sont alors facilement identifiables grâce à la connaissance des niveaux qualitatifs de chaque entité. Aucun des multiplexes n'agit sur son entité cible, les célérités sont donc  $C_{v_1, \emptyset, 1}$  et  $C_{v_2, \emptyset, 0}$ .

D'après la définition de la plus faible précondition, la condition hybride  $H_3$  est obtenue par la formule suivante :

$$\begin{aligned} H_3 &\equiv H_4[\eta_{v_1} \setminus \eta_{v_1} - 1] \wedge \Phi_{v_1}^-(4.0) \wedge \mathcal{F}(4.0) \wedge \\ &\quad \neg\mathcal{W}_{v_1}^- \wedge \mathcal{A}(4.0, \text{slide}^-(v_2)) \wedge \mathcal{J}_{v_1} \end{aligned}$$

1. <http://www.i3s.unice.fr/~comet/DOCUMENTS/hybridisation.tar.gz>

Après simplification des contraintes obtenues pour chaque sous-propriété, on obtient :

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

On réitère ce même procédé successivement sur les trois chemins élémentaires restants pour calculer les contraintes sur toutes les célérités du système. Les contraintes données en annexe 7.2 proviennent de notre outil qui simplifie et réduit les formules.

Le système de contraintes obtenues comprend 26 variables dont 8 célérités (2 célérités pour chacun des 4 états discrets) définies sur  $\mathbb{R}$  et 18 parties fractionnaires définies sur  $[0, 1]$ . En effet, le chemin a 4 chemins élémentaires ce qui définit 4 couples  $(D_i, H_i)_{i \in [0..3]}$  et un couple final  $(D_4, H_4)$ . Pour chacun des 4 couples non finaux, il y a 2 parties fractionnaires pour l'état hybride d'entrée et 2 autres pour l'état hybride de sortie, et pour le couple final, il y a seulement 2 parties fractionnaires pour l'état hybride de sortie. Les simplifications effectuées à la volée permettent de déterminer 6 parties fractionnaires de sortie qui correspondent aux 4 positions de sortie de chacun des *dpa* et aux 2 glissements imposés par la trace biologique. Au final, le CSP contient 38 contraintes dont 6 affectations, voir annexe 7.2. Sur les exemples traités, notre approche a permis de supprimer les contraintes triviales et les contraintes redondantes qui représentent environ la moitié des contraintes générées.

## 4 Solveur

Afin de résoudre ce problème nous utilisons AbSolute [14] un solveur de contraintes continues basé sur les domaines abstraits. Pour des questions d'efficacité, nous avons ajouté dans le solveur une phase de pré-processing durant laquelle, une réécriture symbolique des contraintes est effectuée.

### 4.1 Réécriture symbolique

Similairement à CPLEX, AbSolute effectue une étape de pré-processing. Cette étape peut permettre de simplifier les contraintes et réduire la taille du problème en éliminant des variables par réécriture. Durant cette étape, seules les contraintes linéaires d'égalité sont considérées. Dans la suite de cette section, le terme égalité désigne une contrainte linéaire d'égalité.

Dans un premier temps, nous identifions toutes les constantes et les conservons dans une liste. Une constante correspond à une égalité unaire. Les

constantes sont remplacées dans tout le CSP par leurs valeurs, et ce processus est répété jusqu'à atteindre un point fixe.

Dans un second temps, les égalités binaires sont supprimées du problème et conservées dans une liste comme des vues de variables [16]. L'une des variables est choisie et remplacée dans tout le CSP par sa réécriture, et des contraintes sur les bornes sont ajoutées. Ce processus est lui aussi répété jusqu'à atteindre un point fixe.

**Exemple 1** *Considérons le CSP avec  $V = \{v_1, v_2, v_3\}$  les variables continues prenant leur valeurs dans  $D = \{D_1 = D_2 = D_3 = [-5, 5]\}$ , et  $C = \{c_1 : v_1 = 2, c_2 : v_1 = v_2 + v_3, c_3 : v_3 = 3 \times (v_2)^2\}$ . On peut voir que  $v_1$  est une constante, on la remplace donc partout par sa valeur. La deuxième contrainte devient une vue, on peut donc remplacer  $v_2$  par  $2 - v_3$ . On obtient donc le CSP suivant :  $V' = \{v_3\}$ ,  $D' = \{D_3 = [-5, 5]\}$  et  $C' = \{c_1 : v_3 = 3 \times (2 - v_3)^2, c_2 : 2 - v_3 \geq -5, c_3 : 2 - v_3 \leq 5\}$ . Sur ce petit exemple, on est passé d'un CSP à trois variables, à un CSP avec une constante, une vue et une variable.*

Reprenons l'exemple donné section 3.3, le CSP associé (annexe 7.2) contient 26 variables et 38 contraintes, après l'étape de réécriture, le CSP contient 16 constantes, 5 vues, 5 variables et 26 contraintes (dont 10 pour les bornes des vues).

Cette étape est transparente pour l'utilisateur, et si elles existent, les solutions sont données pour les variables initiales du problème.

### 4.2 Représentation d'une solution

Dans le continu, une solution est généralement un produit d'intervalles appelé boîte. En utilisant la notion de contracteurs [6], une solution est soit une boîte ne contenant que des solutions du CSP, soit une boîte assez petite (pour un paramètre de précision donné) pouvant contenir une solution du CSP. La figure 4 montre un exemple de solutions d'un problème, les boîtes vertes ne contiennent que des solutions et les boîtes roses peuvent en contenir.

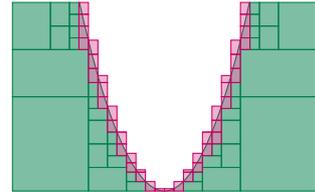


FIGURE 4 – Comparaison entre les solutions sûres et non sûres.

AbSolute peut renvoyer, si elles existent, deux types de solutions : les solutions sûres, ne contenant que des solutions du problème, et les solutions non sûres pouvant en contenir une.

Toute solution est représentée par deux listes. La première contient le domaine des variables sûres (quelque soit la valeur prise dans ce domaine, il existe une solution). Par abus de langage, on parlera de liste d'*invariants*. La seconde liste, quant à elle, contient les domaines pouvant contenir une solution. Par opposition, on parlera de liste de *changeants*.

Une solution sûre aura une liste de changeants vide et toutes les variables seront dans la liste des invariants. Une solution non sûre aura quant à elle au moins un élément dans la liste des changeants.

Lorsqu'une solution du problème est trouvée, les inconnues sont substituées par leur valeur dans les vues. Ces nouvelles affectations sont ajoutées à la liste des invariants ou des changeants suivant que l'inconnue était invariante ou changeante.

## 5 Résultats expérimentaux

### 5.1 Démarche générale

L'utilisation de la logique de Hoare pour contraindre les paramètres du modèle que l'on cherche à construire vise 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 gènes, et d'autre part une postcondition et une trace expérimentale exprimée dans le langage de chemins. Trois principales étapes sont à distinguer : construction des contraintes minimales portant sur les paramètres dynamiques du réseau de gènes, identification d'un jeu de paramètres satisfaisant ces contraintes, et simulation du modèle qui pourra ensuite être comparée à la trace biologique expérimentale.

La première étape consiste à utiliser le calcul de la plus faible précondition à partir de la postcondition donnée en remontant le chemin décrivant la trace biologique observée, voir sous-section 3.2. On se permet, pour les modèles à comportement cyclique, de rajouter des contraintes supplémentaires indiquant que l'état hybride initial du chemin est identique à l'état hybride final. Ces contraintes sont simplifiées et réduites à la volée.

L'étape suivante fait appel au solveur de contraintes AbSolute qui aide à l'identification d'un jeu de paramètres satisfaisant les contraintes. Une solution donnée par le solveur AbSolute correspond à la donnée d'une liste d'invariants et d'une liste de changeants. Si la liste des changeants est vide, 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, on 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.

### 5.2 Résultats

Pour le modèle de la figure 1, à partir du chemin et de la postcondition de la sous-section 3.3, la démarche exposée précédemment mène à un jeu de paramètres que nous avons utilisé pour simuler l'évolution du modèle, voir figure 5. De fait, une solution sûre est immédiatement donnée par AbSolute.

Toute la chaîne de traitement, allant de la construction des contraintes à la simulation est exécutée en 656 millisecondes.

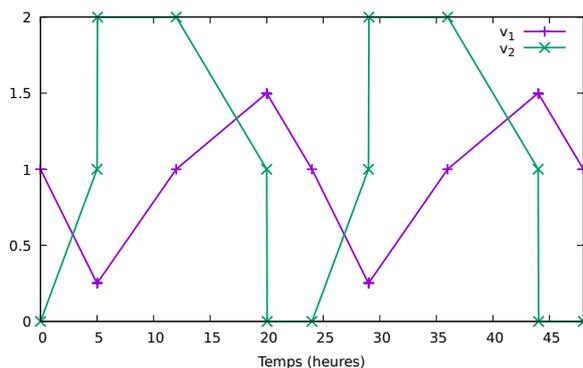


FIGURE 5 – Simulation obtenue sur 48 heures en comportement cyclique. La courbe associée à chacune des entités  $v_i$  correspond à l'évolution de la somme du niveau qualitatif discret et de la partie fractionnaire ( $\eta(v_i) + \pi(v_i)$ ).

Les 4 chemins élémentaires du chemin de la sous-section 3.3 sont facilement identifiables dans la fi-

gure 5 : l'incrémentation du niveau qualitatif de  $v_2$  (resp.  $v_1$ ) observé en ordonnée a lieu à 5.0 heures (resp. 12.0 heures), et sa décrémentation à 20.0 heures (resp. 24.0 heures). Les phénomènes de saturation  $slide^+(v_2)$  et de dégradation totale  $slide^-(v_2)$  sont bien apparents avant le changement de niveau qualitatif  $v_1+$  et  $v_1-$  respectivement. Enfin, l'entité  $v_1$  ne glisse jamais, ce qui est dû à la prise en compte de nos contraintes  $noslide(v_1)$ .

Pour la simulation de la figure 5, nous avons supposé que le système biologique présente un comportement cyclique de 24 heures et cette contrainte a été rajoutée au CSP. On peut voir dans la simulation deux cycles de période de 24 heures. Néanmoins, à chaque période, la phase est décalée de 0.53 secondes et ce décalage est dû à l'approximation des réels dans le solveur. Cependant, cette simulation est robuste puisque le temps nécessaire pour voir la phase se décaler d'une période dépasse 18 ans. Cette précision est largement suffisante pour les problèmes abordés en biologie.

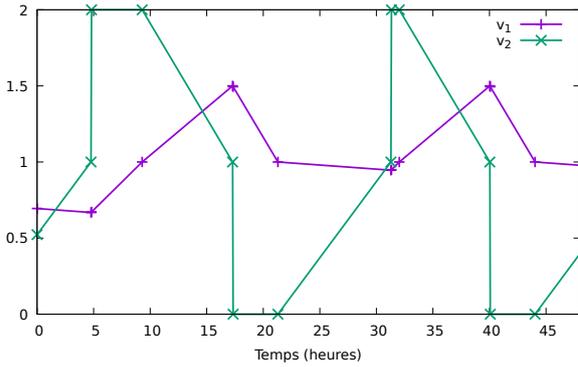


FIGURE 6 – Simulation obtenue sur 48 heures sans imposer un comportement cyclique.

Lorsqu'on n'impose pas de comportement cyclique, la démarche utilisée a du mal à identifier précisément le point initial de la simulation : les variables du CSP correspondant au début de la simulation appartiennent à la liste des changeants. Pour obtenir la figure 6, nous avons choisi au hasard des valeurs pour les parties fractionnaires du point initial, ce qui peut impacter considérablement l'évolution des entités. Néanmoins, les simulations partant d'initialisations différentes mènent après un certain temps à un comportement cyclique. En effet, il a été montré que tout système construit sur un graphe d'interaction de deux variables et dont la dynamique contient un chemin glissant sur un mur externe possède un comportement cyclique appelé *cycle limite* vers lequel toutes les traces convergent [8].

Intuitivement, la figure 7 montre que, lorsqu'on démarre de l'état hybride  $h_3$ , le bord supérieur de  $v_2$  est atteint et franchi, puis une nouvelle transition continue

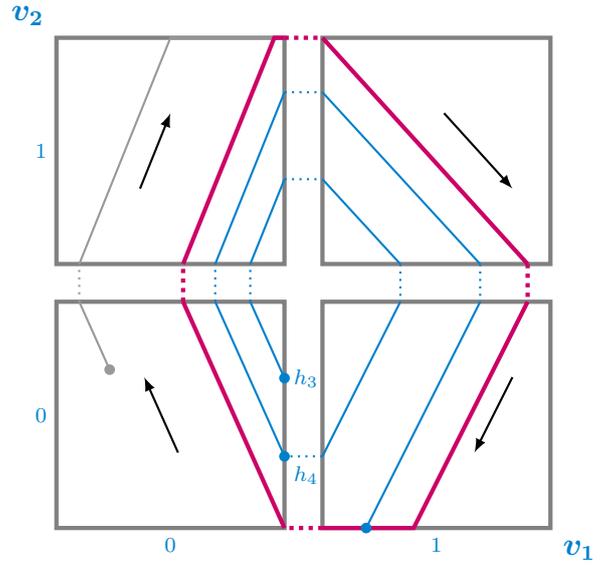


FIGURE 7 – Convergence de trajectoires vers un cycle limite possible associé au graphe de la figure 1. Le cycle limite est représenté en rose, et les traces grises et bleues montrent la convergence des traces vers le cycle limite.

est observée dans l'état discret  $(0, 1)$ . Cette transition continue parvient au bord supérieur de  $v_1$  et l'alternance entre transitions continues et discrètes se poursuit jusqu'à atteindre l'état hybride  $h_4$ . La trajectoire s'est rapprochée du cycle limite en rose, et après plusieurs passages par l'état discret  $(0, 0)$ , le cycle limite est atteint.

## 6 Conclusion et Perspectives

Dans cet article, nous avons couplé le calcul de la plus faible précondition de la logique de Hoare avec le solveur de contraintes AbSolute. Cette approche nous a permis de générer des jeux de paramètres sur des modèles hybrides de réseaux de gènes menant à une dynamique exhibant la trace biologique observée. L'outil développé est très prometteur du point de vue de l'application biologique puisqu'il permet de générer sur nos premiers exemples des comportements similaires aux observations biologiques. Au cours de nos développements, nous avons été confronté à la taille des formules construites. L'explosion combinatoire nous a incité à trouver une stratégie de simplification et de réduction à la volée des contraintes du CSP. De plus, la prise en compte de relations symboliques entre les variables dans le solveur nous a permis de diminuer le nombre de variables à identifier et donc limiter les problèmes d'incertitude entre variables dépendantes dus

à la représentation des nombres réels.

Le temps d'exécution du solveur AbSolute dépend d'une part de la précision sur les intervalles des variables du CSP, et d'autre part du nombre de solutions que l'utilisateur souhaite. Pour aborder des problèmes de plus grandes tailles, notamment les systèmes biologiques complexes, il sera nécessaire de trouver le meilleur compromis entre diversité des solutions et temps de calcul pour obtenir des résultats satisfaisants. Enfin, notre objectif actuel avec retombée thérapeutique est de modéliser le couplage du cycle circadien (alternance sommeil/éveil) avec le cycle cellulaire (division des cellules) pour aborder des questions de chronothérapie du cancer (optimisation de l'heure d'administration des médicaments). Toutefois, le modèle du cycle cellulaire (5 entités biologiques abstraites) mène à un système de contraintes contenant 178 variables continues, 65 variables discrètes, 267 contraintes dont certaines disjonctions ce qui augmente considérablement la combinatoire. Le couplage de ces deux cycles mènera à un système de contraintes qui nécessitera de revoir notre stratégie de recherche de solutions. Notamment, une amélioration du système de réécriture des contraintes continues devrait permettre une résolution plus rapide.

## Références

- [1] R. Alur. Formal verification of hybrid systems. In *2011 Proceedings of the Ninth ACM International Conference on Embedded Software (EMSOFT)*, pages 273–278, Oct 2011.
- [2] J. Behaegel, J.-P. Comet, and M. Folschette. 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)*, pages 5 :1–5 :21, October 2017.
- [3] G. Bernot, J.-P. Comet, Z. Khalis, A. Richard, and O. F. Roux. A genetically modified Hoare logic. *Theoretical Computer Science*, 2018. <https://doi.org/10.1016/j.tcs.2018.02.003>.
- [4] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. Application of formal methods to biological regulatory networks : extending Thomas' asynchronous logical approach with temporal logic. *Journal of theoretical biology*, 229(3) :339–347, 2004.
- [5] A. Bockmayr and A. Courtois. Using hybrid concurrent constraint programming to model dynamic biological systems. In *International Conference on Logic Programming*, pages 85–99. Springer, 2002.
- [6] G. Chabert and L. Jaulin. Contractor programming. *Artificial Intelligence*, 173 :1079–1100, 2009.
- [7] F. Corblin, E. Fanchon, and L. Trilling. Applications of a formal approach to decipher discrete genetic networks. *BMC Bioinformatics*, 11 :385, 2010.
- [8] E. Cornillon. *Modèles qualitatifs de réseaux génétiques : réduction de modèles et introduction d'un temps continu*. PhD thesis, Université Côte d'Azur, 2017.
- [9] E. Dijkstra. Guarded commands, nondeterminacy and formal derivation of programs. *Commun. ACM*, 18 :453–457, August 1975.
- [10] L. F. Fitime, O. F. Roux, C. Guziolowski, and L. Paulevé. Identification of bifurcation transitions in biological regulatory networks using answer-set programming. *Algorithms for Molecular Biology*, 12(1) :19 :1–19 :14, 2017.
- [11] A. Girard, A. A. Julius, and G. J. Pappas. Approximate simulation relations for hybrid systems. *Discrete Event Dynamic Systems*, 18(2) :163–179, 2008.
- [12] T. A. Henzinger. *The Theory of Hybrid Automata*, pages 265–292. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.
- [13] M. Martin, P. Dague, S. Pérès, and L. Simon. Minimality of metabolic flux modes under boolean regulation constraints. In *12th International Workshop on Constraint-Based Methods for Bioinformatics*, 2016.
- [14] M. Pelleau, A. Miné, C. Truchet, and F. Benhamou. A constraint solver based on abstract domains. In *International Conference on Verification, Model Checking, and Abstract Interpretation*, pages 434–454. Springer, 2013.
- [15] G. Rodrigo, J. Carrera, and A. Jaramillo. Combinatorial optimisation to design gene regulatory networks. In *4th International Workshop on Constraint-Based Methods for Bioinformatics*, 2008.
- [16] C. Schulte and G. Tack. View-based propagator derivation - (extended abstract). In *20th International Conference on Principles and Practice of Constraint Programming*, pages 938–942, 2014.
- [17] R. Thomas. Boolean formalization of genetic control circuits. *Journal of theoretical biology*, 42(3) :563–585, 1973.

## 7 Annexe

### 7.1 Sous-propriétés

Nous détaillons ici chacune des sous-propriétés de la plus faible précondition, voir définition 3.

#### 7.1.1 Ressource d'une entité

Pour calculer la plus faible précondition, nous étiquetons les parties fractionnaires des états du système. Nous utilisons les étiquettes  $i$  (initial) et  $f$  (final). De plus, nous utilisons  $\pi'$  (resp.  $\pi$ ) pour spécifier la partie fractionnaire de sortie (resp. d'entrée) de l'état discret courant.

Toutes les propriétés suivantes dépendent des indices  $i$  et  $f$  utilisés dans la définition 3. Pour alléger les notations, les parties fractionnaires  $\pi_v^i$  et  $\pi_v^f$  seront notées simplement  $\pi_v$  et  $\pi'_v$ . Enfin, pour toute entité  $v \in V$  et pour tout sous-ensemble prédécesseur de  $v$   $\omega \subset R^-(v)$ , nous définissons :

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

En d'autres termes,  $\Phi_v^\omega$  est vraie dans un état  $h$  si et seulement si les ressources de  $v$  sont exactement  $\omega$ .

#### 7.1.2 Transition discrète

Pour l'entité du  $dpa$   $v \in V$ ,  $\Phi_v^+(\Delta t)$  (resp.  $\Phi_v^-(\Delta t)$ ) donne les conditions pour que  $v$  incrémente (resp. décrémente) son niveau d'expression discret après  $\Delta t$  unités de temps : sa célérité dans l'état courant est positive (resp. négative) et sa partie fractionnaire d'entrée dépend de sa partie fractionnaire de sortie, de  $\Delta t$  et de  $C_{v,\rho(h,v),n}$ .

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

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

#### 7.1.3 Murs internes et externes

Pour toute entité  $u \in V$ ,  $\mathcal{W}_u^+$  (resp.  $\mathcal{W}_u^-$ ) empêche  $u$  d'incrémenter (resp. décrémente) son niveau qualitatif. Ce mur peut être externe  $\text{EW}_u^+$  (resp.  $\text{EW}_u^-$ ) ou interne  $\text{IW}_u^+$  (resp.  $\text{IW}_u^-$ ). De plus,  $\Phi_{u+}^{\omega'}$  (resp.  $\Phi_{u-}^{\omega'}$ ), requis dans ces sous-formules, est vraie si et seulement

si les ressources de  $u$  sont exactement  $\omega'$  dans l'état où le niveau qualitatif de  $u$  a été incrémenté (resp. décrémente) de 1.

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

avec :

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

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

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

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

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

#### 7.1.4 Entités de *first*

$\mathcal{F}(\Delta t)$  contraint les entités qui ne sont pas dans *first* à atteindre leur bord après les entités de *first*, ou à faire face à un mur interne ou externe.

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

#### 7.1.5 Assertions hybrides

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

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

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

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

Les trois premières sous-formules indiquent que la position de sortie de l'entité  $u$  est située sur un bord. De plus, les contraintes  $\pi_u^i > \pi_u^{i'} - C_{u, \omega, n} \cdot \Delta t$  et  $\pi_u^i < \pi_u^{i'} - C_{u, \omega, n} \cdot \Delta t$  indiquent que la durée avant d'atteindre le bord de  $u$  est inférieure au temps passé dans l'état courant. Le signe de la célérité de l'entité glissante  $u$  est contrainte par la sous-propriété  $\mathcal{F}$  et par la contrainte  $\pi_u > \pi'_u - C_{u, \omega, n} \cdot \Delta t$  (resp.  $\pi_u < \pi'_u - C_{u, \omega, n} \cdot \Delta t$ ) de la sous-propriété  $\mathcal{S}_{u, \omega, n}^+(\Delta t)$  (resp.  $\mathcal{S}_{u, \omega, n}^-(\Delta t)$ ) ou  $\mathcal{S}_{u, \omega, n}(\Delta t)$ .

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

### 7.1.6 Jonctions

**Jonctions continues** Pour toute entité  $v \in V$ , et pour une transition continue entre deux états hybrides  $h = (\eta, \pi)$  et  $h' = (\eta, \pi')$ ,  $\mathcal{C}\mathcal{J}_v$  établit une relation entre les parties fractionnaires et la célérité de l'entité  $v$ . Si la partie fractionnaire de sortie de  $v$  est 0 ou 1, le signe

de la célérité peut être déduit et le temps nécessaire à  $v$  pour atteindre son bord est plus faible que le temps passé dans l'état discret courant. Si  $v$  n'atteint pas son bord, la position exacte de la partie fractionnaire d'entrée de  $v$  peut être déduite de la position de sortie, du temps passé dans l'état discret courant et de la célérité.

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

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

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

Finalement, nous définissons :

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

Ces relations sont facilement interprétables sur la figure 3 : toutes les parties fractionnaires restent les mêmes entre les deux états discrets sauf la partie fractionnaire de l'entité assurant la transition.

## 7.2 Contraintes du système

Après simplification et réduction des contraintes construites sur l'exemple donné en section 3.3, en y ajoutant un comportement cyclique, nous obtenons 6 valeurs exactes sur des parties fractionnaires, ainsi que les 32 contraintes suivantes :

$$\begin{aligned} &(\pi_{v_1}^{3'} = 0.) \wedge (\pi_{v_2}^{3'} = 0.) \wedge (\pi_{v_2}^{2'} = 0.) \wedge (\pi_{v_1}^{1'} = 1.) \wedge \\ &(\pi_{v_2}^{1'} = 1.) \wedge (\pi_{v_2}^{0'} = 1.) \wedge (\pi_{v_1}^4 = \pi_{v_1}^0) \wedge (\pi_{v_2}^4 = \pi_{v_2}^0) \wedge \\ &(C_{v_1, \emptyset, 0} \times C_{v_1, \emptyset, 1} \geq 0.) \wedge (C_{v_1, \{m2\}, 0} \times C_{v_1, \{m2\}, 1} \geq \\ &0.) \wedge (C_{v_2, \emptyset, 0} \times C_{v_2, \emptyset, 1} \geq 0.) \wedge (C_{v_2, \{m1\}, 0} \times C_{v_2, \{m1\}, 1} \geq \\ &0.) \wedge (\pi_{v_1}^{3'} = 1. - \pi_{v_1}^4) \wedge (\pi_{v_2}^{3'} = \pi_{v_2}^4) \wedge (\pi_{v_1}^{2'} < \\ &1.) \wedge (\pi_{v_1}^{2'} > 0.) \wedge (\pi_{v_2}^{2'} = 1. - \pi_{v_2}^3) \wedge (\pi_{v_1}^{2'} = \pi_{v_1}^3) \wedge \\ &(\pi_{v_1}^{1'} = 1. - \pi_{v_1}^2) \wedge (\pi_{v_2}^{1'} = \pi_{v_2}^2) \wedge (\pi_{v_1}^{0'} < 1.) \wedge (\pi_{v_1}^{0'} > \\ &0.) \wedge (\pi_{v_2}^{0'} = 1. - \pi_{v_2}^1) \wedge (\pi_{v_1}^{0'} = \pi_{v_1}^1) \wedge (\pi_{v_2}^0 = \\ &\pi_{v_2}^{0'} - C_{v_2, \{m1\}, 0} \times 5.) \wedge (C_{v_2, \{m1\}, 0} > 0.) \wedge (\pi_{v_1}^0 = \\ &\pi_{v_1}^{0'} - C_{v_1, \emptyset, 0} \times 5.) \wedge (C_{v_2, \{m1\}, 1} > 0.) \wedge (\pi_{v_1}^1 = \end{aligned}$$

$\pi_{v_1}^{1'} - C_{v_1, \{m_2\}, 0} \times 7.) \wedge (C_{v_1, \{m_2\}, 0} > 0.) \wedge (\pi_{v_2}^2 = \pi_{v_2}^{2'} - C_{v_2, \emptyset, 1} \times 8.) \wedge (C_{v_2, \emptyset, 1} < 0.) \wedge (\pi_{v_1}^2 = \pi_{v_1}^{2'} - C_{v_1, \{m_2\}, 1} \times 8.) \wedge (C_{v_2, \emptyset, 0} < 0.) \wedge (\pi_{v_1}^3 = \pi_{v_1}^{3'} - C_{v_1, \emptyset, 1} \times 4.) \wedge (C_{v_1, \emptyset, 1} < 0.) \wedge (\pi_{v_2}^1 > \pi_{v_2}^{1'} - C_{v_2, \{m_1\}, 1} \times 7.) \wedge (\pi_{v_2}^3 < \pi_{v_2}^{3'} - C_{v_2, \emptyset, 0} \times 4.);$  où  $m_1$  représente le multiplexe actif relatif à la formule  $\neg(v_1 \geq 1)$  et  $m_2$  celui relié à la formule  $v_2 \geq 1$ .