



HAL
open science

Réseaux booléens : formalisme, sémantiques, et complexité

Loïc Paulevé

► **To cite this version:**

Loïc Paulevé. Réseaux booléens : formalisme, sémantiques, et complexité. Cédric Lhoussaine; Élisabeth Remy. *Approches symboliques de la modélisation et de l'analyse des systèmes biologiques*, ISTE Editions, 2022, 9781789490299. hal-03612593

HAL Id: hal-03612593

<https://hal.science/hal-03612593v1>

Submitted on 21 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Réseaux booléens : formalisme, sémantiques, et complexité

Loïc PAULEVÉ

CNRS, LaBRI UMR58000, Talence, France

Ce chapitre présente le formalisme des réseaux booléens et leur relation avec des modèles classiques issus de l'informatique fondamentale, en particulier avec les réseaux de Petri avec différentes sémantiques de mise à jour. Motivé par la modélisation des systèmes biologiques, ce chapitre s'attarde alors sur la sémantique la plus permissive (MP) des réseaux booléens qui garantit de capturer tout comportement possible avec tout modèle plus quantitatif respectant la logique du réseau. Le chapitre conclut sur la complexité de décision de propriétés dynamiques élémentaires selon les différentes sémantiques abordées, montrant là encore une différence notable entre les sémantiques classiques de mise à jour et la sémantique MP.

0.1. Introduction

La modélisation mathématique et informatique peut aider à comprendre et *in fine* à contrôler le comportement du vivant. Par exemple au niveau de la cellule, en modélisant les interactions connues entre des gènes et protéines il est possible d'identifier, par simulations ou par raisonnement logique, des cibles potentielles pour contrôler, positivement ou négativement, l'occurrence de phénotypes d'intérêts (Saez-Rodriguez *et al.* 2009, Le Novère 2015, Collombet *et al.* 2017, Zañudo *et al.* 2018). Ce type de modèle est souvent qualifié de "mécanistique", car il cherche à reproduire les interactions physico-chimiques entre des entités clairement identifiées, toutefois toujours avec un certain degré de simplification.

Les réseaux booléens (Kauffman 1969, Thomas 1973) sont de plus en plus employés pour modéliser les réseaux de régulation génétique et les voies de

signalisation en tenant compte d'un grand nombre d'entités (p.ex. Cohen *et al.* 2015, Abou-Jaoudé *et al.* 2015). L'activité (ou état) des entités sélectionnées est simplifiée de manière binaire : inactif/actif, absent/présent, 0/1, *etc.* L'évolution de l'activité de chaque entité est alors décrite par des règles logiques qui dépendent de l'état de ses régulateurs. Ceci reflète en partie le type de connaissance généralement disponible pour la modélisation des systèmes biologiques, centré sur l'existence d'interactions entre gènes et protéines mais avec peu de détails sur les concentrations et cinétiques. Ainsi, les modèles booléens se veulent être une simplification à gros grains du système étudié, mais robuste car ils ne dépendent que de peu de paramètres, et en particulier n'ont aucun paramètre à domaine continu.

Un réseau booléen est la donnée d'une fonction des vecteurs binaires de dimension n vers eux-mêmes, $f : \mathbb{B}^n \rightarrow \mathbb{B}^n$ avec $\mathbb{B} = \{0, 1\}$, et d'une sémantique qui spécifie les évolutions possibles d'un vecteur binaire $x \in \mathbb{B}^n$ sachant que $f(x) = x'$. Étant donné $i \in \{1, \dots, n\}$, on note $f_i : \mathbb{B}^n \rightarrow \mathbb{B}$ la i -ème composante de f , aussi appelée fonction locale de i ; ainsi $f(x) = (f_1(x), \dots, f_n(x))$.

EXEMPLE.– Avec la fonction f de dimension 3 où

$$f_1(x) = x_3 \wedge (\neg x_1 \vee \neg x_2)$$

$$f_2(x) = x_3 \wedge x_1$$

$$f_3(x) = x_1 \vee x_2 \vee x_3 ,$$

$$f(011) = (f_1(011), f_2(011), f_3(011)) = 101, f(100) = 001, \text{ etc.}$$

Nous appelons *configuration* du réseau les vecteurs binaires de dimension n . Une configuration modélise l'état de tous les composants du réseau. Pour les applications en biologie, ces composantes peuvent faire référence à des gènes, protéines, ARN, *etc.*, mais aussi des notions plus abstraites comme des phénotypes (apoptose, arrêt du cycle cellulaire, *etc.*) ; avec selon les cas différentes interprétations biologiques de l'état 0 et 1 de ces composantes.

Intuitivement, chaque f_i modélise les conditions nécessaires et suffisantes pour activer (état 1) le i -ème composant du réseau, en fonction de l'état de tous les composants du réseau. Nous verrons dans la section suivante différentes manières d'interpréter ce f (sémantiques) pour calculer la *dynamique* des configurations d'un réseau booléen.

Dans les applications aux systèmes biologiques, il est courant que la fonction f soit *localement monotone* : pour chaque $i \in \{1, \dots, n\}$, f_i dépend soit toujours positivement, soit toujours négativement, soit aucunement de chaque composante $j \in \{1, \dots, n\}$. Ceci traduit le fait qu'un composant du réseau ne peut agir à la fois comme un activateur et un inhibiteur d'un même composant. Ceci exclut par exemple les

fonctions du type $f_1(x) = x_2 \text{ xor } x_3$, où la croissance de x_2 peut entraîner à la fois une croissance et une décroissance de $f_1(x)$ selon la valeur de x_3 . À noter que la monotonie locale n'implique pas la monotonie de f . Cette classe de fonctions peut se caractériser formellement ainsi :

DÉFINITION.– La fonction $f : \mathbb{B}^n \rightarrow \mathbb{B}^n$ est *localement monotone* si pour chaque composante $i \in \{1, \dots, n\}$, il existe un ordre par composantes $\preceq^i \in \{\leq, \geq\}^n$ tel que

$$\forall x, y \in \mathbb{B}^n, \quad (x_1 \preceq_1^i y_1 \wedge \dots \wedge x_n \preceq_n^i y_n) \Rightarrow f_i(x) \leq f_i(y) .$$

Le réseau booléen donné précédemment en exemple est localement monotone, par exemple avec $\preceq^1 = (\geq, \geq, \leq)$ et $\preceq^2 = \preceq^3 = (\leq, \leq, \leq)$.

Ce chapitre aborde quelques aspects formels de la modélisation et l'analyse des réseaux booléens.

En premier lieu, les sémantiques classiquement utilisées pour interpréter la fonction d'un réseau booléen sont définies dans la Section 0.2, avec quelques remarques sur leurs relations. Les liens entre les réseaux booléens et différents formalismes informatiques classiques sont discutés dans la Section 0.3, avec un focus sur les réseaux de Petri, en montrant une traduction bidirectionnelle.

Les réseaux booléens sont communément employés pour modéliser des systèmes dont l'état des composants ne sont pas binaires. Dans la Section 0.4, nous étudions certaines garanties sur les propriétés obtenues sur un réseau booléen vis-à-vis de modèles plus quantitatifs, tels que les réseaux multivalués, ou encore les systèmes à équations différentielles ordinaires. Il apparaît que les sémantiques classiques des réseaux booléens n'offrent *aucune* garantie sur l'existence ou l'absence de trajectoires dans ces modèles plus concrets. Nous proposons alors une nouvelle interprétation des réseaux booléens, les réseaux booléens MP (*most permissive Boolean networks*), qui apporte des garanties formelles très fortes sur les comportements prédits. En particulier, si un comportement est impossible avec cette interprétation, il est impossible pour tout modèle quantitatif qui respecte la logique du réseau booléen.

Enfin, les complexités pour la décision de différentes propriétés dynamiques usuelles (points fixes, accessibilité, attracteurs) sont données dans la Section 0.5 pour les sémantiques classiques et pour la MP. Il en résulte que l'analyse des réseaux booléens MP est sensiblement moins complexe, et est ainsi applicable à des réseaux à l'échelle du génome, tout en offrant des garanties vis-à-vis des comportements de modèles issus de raffinements quantitatifs.

Notations

Dans la suite de ce chapitre, f désigne la fonction d'un réseau booléen de dimension n . L'ensemble des composantes $\{1, \dots, n\}$ est abrégé en $[n]$. Étant

données deux configurations $x, y \in \mathbb{B}^n$, les composantes qui diffèrent sont dénotées par $\Delta(x, y) \triangleq \{i \in [n] \mid x_i \neq y_i\}$. Le symbole \wedge dénote la conjonction, \vee la disjonction et \neg la négation. Étant donné un ensemble fini S , $|S|$ est sa cardinalité.

0.2. Sémantiques classiques des réseaux booléens

La fonction $f : \mathbb{B}^n \rightarrow \mathbb{B}^n$ d'un réseau booléen spécifie l'état vers lequel tend chaque composant dans chaque configuration. La sémantique définit alors l'évolution des configurations en accord avec f . Historiquement, deux sémantiques sont principalement usitées : la sémantique de mise à jour synchrone où tous les composants changent d'état simultanément (Kauffman 1969), et la sémantique pleinement asynchrone (Thomas 1973), où seul un composant, choisi de manière non déterministe, peut changer d'état. De nombreuses autres variantes ont été introduites et étudiées dans la littérature, comme les sémantiques séquentielles et bloc-séquentielles, où est spécifié l'ordre des composants à mettre à jour (Robert 1995, Aracena *et al.* 2009). Dans ce chapitre, nous considérons dans un premier temps la sémantique synchrone, pleinement asynchrone, et asynchrone, cette dernière englobant la majorité, si ce n'est toutes des sémantiques de mise à jour introduites dans la littérature des réseaux booléens.

0.2.1. Définitions

Les sémantiques de mise à jour sont définies ici comme des relations binaires irréflexives entre les configurations du réseau. Nous utilisons le symbole \rightarrow décoré de f et d'un symbole représentant la sémantique employée.

DÉFINITION.– *Mise à jour synchrone*

$$\forall x, y \in \mathbb{B}^n \quad x \xrightarrow[s]{f} y \iff x \neq y \wedge y = f(x) . \quad [1]$$

DÉFINITION.– *Mise à jour pleinement asynchrone*¹

$$\forall x, y \in \mathbb{B}^n, \quad x \xrightarrow[a]{f} y \iff \exists i \in [n] : \Delta(x, y) = \{i\} \wedge y_i = f_i(x) . \quad [2]$$

1. Dans la littérature des réseaux booléens, le *pleinement asynchrone* est souvent appelé (seulement) *asynchrone*, alors que l'*asynchrone*, peu étudié, est souvent appelé *asynchrone généralisé*. Nous adoptons dans ce chapitre la terminologie classique pour les systèmes dynamiques discrets (Manzoni 2012, Fatès 2018).

DÉFINITION.– *Mise à jour asynchrone*

$$\forall x, y \in \mathbb{B}^n, \quad x \xrightarrow[\text{a}^*]{f} y \triangleq x \neq y \wedge \forall i \in \Delta(x, y), y_i = f_i(x) . \quad [3]$$

Étant donnée une sémantique σ , on écrit $\xrightarrow[\sigma]{f}^*$ la fermeture réflexive et transitive de la relation binaire $\xrightarrow[\sigma]{f}$. Il en résulte que $x \xrightarrow[\sigma]{f}^* y$ si et seulement si $x = y$ ou il existe une séquence $x \xrightarrow[\sigma]{f} x' \xrightarrow[\sigma]{f} \dots \xrightarrow[\sigma]{f} y$. L'ensemble des configurations en relation transitive avec une configuration x est donné par $\rho_\sigma^f(x)$:

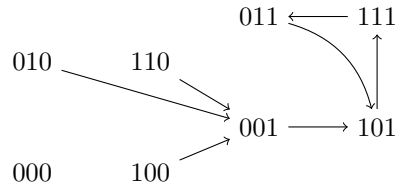
$$\rho_\sigma^f(x) \triangleq \{y \in \mathbb{B}^n \mid x \xrightarrow[\sigma]{f}^* y\} . \quad [4]$$

0.2.2. Exemples

Considérons la fonction f de dimension 3 donnée en exemple p.2.

Sémantique synchrone

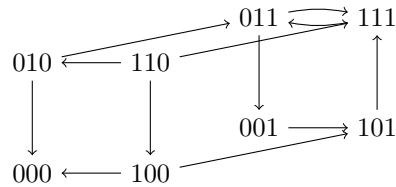
La relation binaire $\xrightarrow[\text{s}]{f}$ est la suivante, où nous omettons les décorations sur les arcs par soucis de lisibilité :



Ainsi, $\rho_s^f(111) = \{111, 011, 101\}$ et $\rho_s^f(010) = \{010, 001, 101, 111, 011\}$.

Sémantique pleinement asynchrone

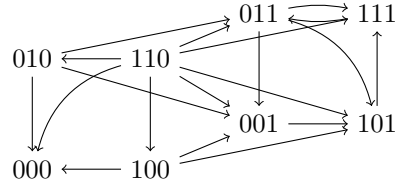
La relation binaire $\xrightarrow[\text{a}]{f}$ est la suivante :



Ici, $\rho_a^f(111) = \{111, 011, 001, 101\}$ et $\rho_a^f(010) = \{010, 000, 011, 001, 101, 111\}$.

Sémantique asynchrone

La relation binaire \xrightarrow{f}_{a^*} est la suivante :



Ici, $\rho_{a^*}^f(111) = \rho_a^f(111)$ et $\rho_{a^*}^f(110) = \rho_a^f(110)$.

0.2.3. Propriétés

Le choix d'une sémantique peut avoir un effet majeur sur les évolutions des configurations d'un réseau booléen. Étant donné une configuration initiale x , l'ensemble des configurations $\rho_\sigma^f(x)$ détermine complètement les futures configurations possibles du réseau. Or, dans le cas général, cet ensemble peut être sensiblement différent selon la sémantique sélectionnée : $\rho_s^f(x) \neq \rho_a^f(x) \neq \rho_{a^*}^f(x)$. Dans l'exemple précédent, la configuration 010 ne peut pas atteindre la configuration 000 selon la sémantique synchrone, contrairement aux sémantiques asynchrones.

Par définition, la mise à jour asynchrone inclut les mises à jour synchrones et pleinement asynchrones. Ainsi, $\xrightarrow{f}_s^* \subseteq \xrightarrow{f}_{a^*}^*$ et $\xrightarrow{f}_a^* \subseteq \xrightarrow{f}_{a^*}^*$, et donc pour toute configuration $x \in \mathbb{B}^n$, $\rho_s^f(x) \subseteq \rho_{a^*}^f(x)$ et $\rho_a^f(x) \subseteq \rho_{a^*}^f(x)$. Dans le cas général, la mise à jour asynchrone peut inclure d'autres relations (il existe des f telles que $\xrightarrow{f}_s^* \cup \xrightarrow{f}_a^* \neq \xrightarrow{f}_{a^*}^*$), dues aux mises à jour simultanées d'entre 2 et $n - 1$ composants.

Noual et Sené (Noual and Sené 2017) ont déterminé une classe de réseaux booléens dits "insensibles" au synchronisme, où toute mise à jour synchrone peut se décomposer en séquences de mises à jour pleinement asynchrones. Leur résultat repose sur le *graphe d'influence* des réseaux booléens, qui résume les dépendances dirigées et signées entre les composants et leurs fonctions. Il résulte que si le graphe d'influence de f est simple (au plus un seul arc entre deux sommets) et qu'il ne contient aucun cycle de longueur paire avec un nombre pair d'arcs négatifs, et aucun cycle de longueur impaire avec un nombre impair d'arcs négatifs, alors f vérifie la propriété suivante : $\forall x \in \mathbb{B}^n$, $\rho_s^f(x) \subseteq \rho_a^f(x)$ et $\rho_a^f(x) = \rho_{a^*}^f(x)$. Une façon

d'interpréter ce résultat est de remarquer que, dans un réseau insensible au synchronisme, la mise à jour synchrone de plusieurs composants ne crée pas de nouveaux comportements par rapport à la mise à jour pleinement asynchrone.

De nombreux autres modes de mise à jour ont été étudiés dans la littérature, tels que les séquentiels et séquentiels par bloc, imposant un ordre (partiel) entre les composants à mettre à jour (Aracena *et al.* 2009). Tous ces modes donnent des relations incluses dans les mises à jour asynchrones.

0.3. Formalismes connexes

0.3.1. Automates cellulaires

Les réseaux booléens appartiennent à la grande famille des systèmes dynamiques discrets. De part leur spécification sous forme de fonction, les réseaux booléens peuvent s'apparenter aux automates cellulaires (Wolfram 1983). Un automate cellulaire est défini sur une grille de cellules selon un nombre de dimensions fixé. La grille est généralement infinie (classiquement \mathbb{Z}^m), mais peut également être finie, par exemple un anneau $\mathbb{Z}/n\mathbb{Z}$. Chaque cellule prend un état parmi un ensemble discret et fini de valeurs ; une configuration associe à chaque cellule un état. Enfin, une fonction dite de voisinage spécifie un ensemble de translations pour obtenir les cellules "voisines" (par exemple les cellules adjacentes) et une (unique) règle locale spécifie le calcul de l'état de chaque cellule en fonction de l'état de son voisinage. Outre la dimension finie des réseaux booléens, leur différence fondamentale par rapport aux automates cellulaires est que chaque composant (cellule) a une fonction de voisinage et une règle locale qui lui est propre. À l'inverse, les automates cellulaires à 2 valeurs en dimension finie peuvent être vus comme un cas particulier de réseaux booléens, où la règle locale est dupliquée pour chaque composant. Tout comme pour les réseaux booléens, différentes sémantiques d'application de la règle locale similaires à celles définies dans la section précédente ont été étudiées (Fatès 2018).

0.3.2. Réseaux de Petri

Les réseaux booléens, tout comme les automates cellulaires, ont une spécification basée sur les fonctions : l'état de chaque composant est calculé par une fonction déterministe, évaluée sur la configuration du réseau. Les réseaux de Petri (Petri 1962, Murata 1989) ont une spécification basée sur les transitions qui listent explicitement les conditions dans lesquelles un ou plusieurs changements d'état ont lieu. Ce type de représentation permet entre autres de raisonner sur la causalité des changements d'états des composants, et d'éviter des explorations redondantes de configurations en exploitant la *concurrency* (indépendance) entre certaines transitions.

De nombreuses variantes de réseaux de Petri ont été considérées dans la littérature. Nous choisissons ici de montrer la relation des réseaux booléens avec les réseaux de Petri *saufs avec arcs de lecture* (*safe Petri nets with read arcs*) (Vogler *et al.* 1998), une variante classique des méthodes formelles autour de la théorie de la concurrence dans les systèmes dynamiques discrets, et dont le lien avec les réseaux booléens est le plus direct.

Un réseau de Petri est un graphe dirigé biparti, composé d'un ensemble de *places* et de *transitions*. Les places peuvent contenir des jetons : les transitions consomment un jeton de chaque place entrante et produisent un jeton dans chaque place sortante. Un réseau de Petri est dit *sauf* si, dans toute exécution, une place reçoit au plus un seul jeton. Une place est alors marquée (avec un jeton), soit non marquée. Le *marquage* d'un réseau de Petri est l'ensemble des places marquées. Dans les réseaux de Petri avec arcs de lecture, chaque transition est associée avec un ensemble de places de pré-condition, post-condition, et de contexte ; par soucis de simplicité, nous considérons que ces ensembles sont disjoints.

DÉFINITION.— Un *réseau de Petri avec arcs de lecture* (RPN ; *Read Petri Net*) est un tuple $(P, T, pre, ctx, post, M_0)$ où P et T sont respectivement des ensembles finis de *places* et de *transitions* ; pre, ctx et $post : T \rightarrow 2^P$ associent chaque transition $t \in T$ à un ensemble de places, tels que $pre(t) \neq \emptyset$, $pre(t) \cap post(t) = \emptyset$, et $ctx(t) \cap (pre(t) \cup post(t)) = \emptyset$; $M_0 \subseteq P$ est le *marquage initial*. Dans la suite, N désigne un RPN $(P, T, pre, ctx, post, M_0)$.

Une transition ne peut être déclenchée que si toutes les places de sa pré-condition et de son contexte sont marquées. L'application d'une transition enlève dans un premier temps le marquage des places de la pré-condition, puis ajoute un marquage sur les places de la post-condition ; les places du contexte restent inchangées. Partant d'un marquage initial, un réseau de Petri est dit *sauf* si dans tout marquage accessible, aucune transition déclenchable n'a de place de sa post-condition marquée.

DÉFINITION.— La *sémantique atomique* d'un RPN N est une relation binaire entre marquages $\xrightarrow[\text{atom}]{N} \subseteq 2^P \times 2^P$ telle que $\forall M, M' \subseteq P$,

$$M \xrightarrow[\text{atom}]{N} M' \stackrel{\Delta}{\iff} \exists t \in T : pre(t) \cup ctx(t) \subseteq M \\ \wedge M' = (M \setminus pre(t)) \cup post(t) .$$

EXEMPLE.— La figure 1 donne la représentation graphique du RPN $N = (P, T, pre, ctx, post, M_0)$ où $P = \{p_1, p_2, p_3, p_4\}$, $T = \{a, b, c\}$, $pre(a) = post(b) = \{p_1\}$, $post(a) = pre(b) = \{p_3\}$, $pre(c) = \{p_2\}$, $post(c) = \{p_4\}$, $ctx(a) = \{p_2\}$, $ctx(b) = ctx(c) = \emptyset$. et $M_0 = \{p_1, p_2\}$.

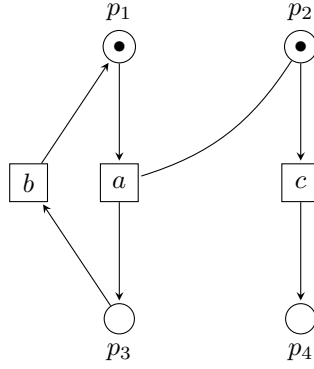


Figure 1. Exemple de RPN sauf. Les places sont représentées par des cercles et les transitions par des carrés ; les pré- et post-conditions par des arcs dirigés, et les contextes par des arcs non-dirigés.

On obtient, entre autres, $\{p_1, p_2\} \xrightarrow[\text{atom}]{N} \{p_3, p_2\} \xrightarrow[\text{atom}]{N} \{p_1, p_2\} \xrightarrow[\text{atom}]{N} \{p_1, p_4\}$.

Nous pouvons déjà effectuer une analogie entre la sémantique atomique des RPN, où une et une seule transition est appliquée à la fois, et la sémantique pleinement asynchrone des réseaux booléens, où un et un seul composant est mise à jour à la fois. Toutefois, il faut bien remarquer d'une transition d'un RPN peut modifier simultanément le marquage de plusieurs places.

Tout comme pour les réseaux booléens, l'application simultanée de transitions est largement étudiée dans la littérature des RPN (Janicki *et al.* 2015, Chatain *et al.* 2015) : on parle ici de sémantiques par pas (*step semantics*). Deux transitions peuvent être appliquées simultanément seulement si l'intersection entre leurs pre-conditions est vide, mais avec potentiellement des places partagées par leurs contextes.

DÉFINITION.— Un ensemble non-vidé de transitions $S \subseteq T$ est un *pas* admissible dans un marquage $M \subseteq P$ si pour tout $t \in S$, $pre(t) \cup ctx(t) \subseteq M$ et pour tout $t, t' \in S$, $pre(t) \cap pre(t') = \emptyset$.

La *sémantique par pas* est une relation binaire entre marquages de $N \xrightarrow[\text{step}]{N} \subseteq 2^P \times 2^P$ telle que $\forall M, M' \subseteq P$,

$$M \xrightarrow[\text{step}]{N} M' \iff \exists S \subseteq T : S \text{ est un pas admissible dans } M$$

$$\wedge M' = (M \setminus \bigcup_{t \in S} pre(t)) \cup \bigcup_{t \in S} post(t) .$$

DÉFINITION.– Un pas S admissible dans un marquage M est maximal si il n'existe aucune transition $t \in T \setminus S$ telle que $S \cup \{t\}$ est un pas admissible dans M .

La *sémantique par pas maximal* est une relation binaire entre marquages de N
 $\xrightarrow[\text{mstep}]{N} \subseteq 2^P \times 2^P$ telle que $\forall M, M' \subseteq P$,

$$M \xrightarrow[\text{mstep}]{N} M' \iff \exists S \subseteq T : S \text{ est un pas } \textit{maximal} \text{ admissible dans } M \\ \wedge M' = (M \setminus \bigcup_{t \in S} \textit{pre}(t)) \cup \bigcup_{t \in S} \textit{post}(t) .$$

La sémantique par pas est analogue à la sémantique asynchrone des réseaux booléens, et la sémantique par pas maximal à la sémantique synchrone.

NOTE.– Il est important de remarquer que pour un RPN N sauf et sans arc de lecture (pour toute transition $t \in T$, $ctx(t) = \emptyset$), ces trois sémantiques (atomique, par pas, par pas maximal) sont similaires : toute application simultanée de transitions peut se décomposer en séquences d'applications de transition avec la sémantique atomique. Ainsi, un RPN sauf sans arc de lecture est insensible au synchronisme : $\rho_{\text{mstep}}^N(M_0) \subseteq \rho_{\text{atom}}^N(M_0) = \rho_{\text{step}}^N(M_0)$.

0.3.2.1. Des réseaux booléens aux réseaux de Petri...

Les réseaux booléens peuvent se traduire en RPN saufs où chaque composant $i \in [n]$ est modélisé par deux places, une pour chaque valeur possible du composant, et où les transitions reflètent l'ensemble des conditions (modélisés par des arcs de lecture) pour les changements de valeurs suivant f_i . Ces changements de valeurs sont possibles dans toute configuration $x \in \mathbb{B}^n$ où $x_i \neq f_i(x)$. Les conditions minimales pour ces changements sont obtenus symboliquement avec les impliquants premiers² (*prime implicants*) des formules en logique propositionnelle du premier ordre $\neg x_i \wedge f_i(x)$ et $x_i \wedge \neg f_i(x)$. Il est à noter que cela peut résulter en un nombre de transitions exponentiel avec le nombre de composants du réseau.

L'encodage décrit ici est similaire à celui initialement proposé par Chaouiya *et al.* (2004), hormis l'emploi d'arcs de lecture qui permet d'exploiter plus précisément les notions de concurrence et de sensibilité au synchronisme. Cette traduction résulte en une classe particulière de RPN saufs où à toute place p correspond une unique place distincte \bar{p} (complément) telles que tout marquage accessible contient soit p soit \bar{p} et jamais les deux ; et où toute transition a exactement une seule place en pré-condition et son complément en post-condition (avec $\bar{\bar{p}} = p$).

2. conjonctions de littéraux minimales (au sens de l'inclusion de l'ensemble des littéraux) impliquant la formule logique

On note $\text{PI}[F]$ les impliquants premiers de la formule F , qui donnent un ensemble de clauses normales conjonctives. Un impliquant $C \in \text{PI}[F]$ est représenté par un ensemble de littéraux, positifs ou négatifs. Pour chaque composant $i \in [n]$, deux places modélisant les états possibles de i sont créées : une place i pour l'état 0, et une place $i + n$ pour l'état 1.

DÉFINITION.– (Chatain et al. 2019) Étant donné un réseau booléen f de dimension n et une configuration $x^0 \in \mathbb{B}^n$, $\langle f \rangle$ est le RPN $(P, T, pre, ctx, post, M_0)$ tel que

- $P = \{1, \dots, 2n\}$ sont les places ;
- T est le plus petit ensemble tel que pour tout $i \in [n]$, pour chaque impliquant $C \in \text{PI}[\neg x_i \wedge f_i(x)]$ (resp. $C \in \text{PI}[x_i \wedge \neg f_i(x)]$), il existe une transition $t \in T$ telle que $pre(t) = \{i\}$ et $post(t) = \{i + n\}$ (resp. $pre(t) = \{i + n\}$ et $post(t) = \{i\}$), et $ctx(t) = \{j \mid \neg x_j'' \in C, j \neq i\} \cup \{j + n \mid x_j'' \in C, j \neq i\}$;
- $M_0 = \langle x^0 \rangle$, où, pour toute configuration $x \in \mathbb{B}^n$, $\langle x \rangle \triangleq \{i + nx_i \mid i \in [n]\}$ (p.ex., $\langle 010 \rangle = \{1, 5, 3\}$, $\langle 101 \rangle = \{4, 2, 6\}$).

La figure 2 donne un exemple de RPN résultant de cet encodage.

On obtient alors une relation de bisimilarité entre tout réseau booléen et son encodage en réseau de Petri en considérant

- la sémantique pleinement asynchrone et atomique ;
- synchrone et par pas maximal,
- asynchrone et par pas :

THÉORÈME.– (Chatain et al. 2019) Étant donné un réseau booléen f de dimension n , pour toute configuration $x, y \in \mathbb{B}^n$,

$$x \xrightarrow[\text{a}]{f} y \iff \langle x \rangle \xrightarrow[\text{atom}]{\langle f \rangle} \langle y \rangle ,$$

$$x \xrightarrow[\text{s}]{f} y \iff \langle x \rangle \xrightarrow[\text{mstep}]{\langle f \rangle} \langle y \rangle ,$$

$$x \xrightarrow[\text{a*}]{f} y \iff \langle x \rangle \xrightarrow[\text{step}]{\langle f \rangle} \langle y \rangle .$$

0.3.2.2. ... et vice versa

Tout RPN $N = (P, T, pre, ctx, post, M_0)$ sauf peut également se traduire en réseau booléen, avec une complexité linéaire.

La principale difficulté réside dans le fait qu'une transition d'un RPN peut modifier simultanément le marquage de plusieurs places, même avec la sémantique

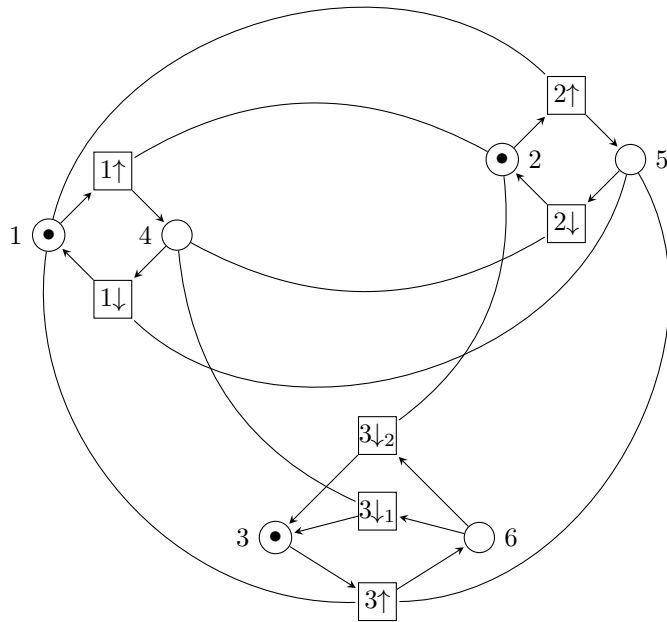


Figure 2. RPN correspondant au réseau booléen f de dimension 3 avec $f_1(x) = \neg x_2$, $f_2(x) = \neg x_1$, $f_3(x) = \neg x_1 \wedge x_2$ et la configuration initiale 000. Les transitions sont notées $i\uparrow$ et $i\downarrow$ pour le changement d'état du composant i de 0 vers 1 et inversement, avec un indice dans les cas où plusieurs impliquants sont nécessaires :
 $PI[x_3 \wedge \neg f_3(x)] = \{\{x_1\}, \{\neg x_2\}\}$, d'où les transitions $3\downarrow_1$ et $3\downarrow_2$ correspondant aux impliquants x_1 (place 4) et $\neg x_2$ (place 2).

atomique. L'idée est alors d'associer un composant du réseau à chaque place et à chaque transition du RPN et de décomposer l'application d'une transition t en plusieurs étapes : quand les composants correspondant à la pré-conditions et au contexte d'une transition ont tous la valeur 1, et qu'aucun composant correspondant à une transition n'est dans l'état 1, le composant correspondant à t est mis à 1. Puis, les composants pour les places de pré- et post-conditions sont mis à jour (dans n'importe quel ordre) pour refléter leur présence dans le marquage final. Enfin, le composant pour t est mis à 0. Alors, la fonction modélisant le marquage d'une place p s'évalue à 1 si et seulement si une transition où p est une post-condition est en cours ; ou p est marquée et aucune transition où p est une pré-condition n'est en cours. La fonction modélisant l'application d'une transition t s'évalue à 1 si et seulement si aucune

transition n'est en cours et toutes les places de la pré-condition et du contexte sont marquées ; ou la transition t est en cours et soit une place de la post-condition n'est pas encore marquée soit une place de la pré-condition est encore marquée.

Il est à noter que la traduction, et plus particulièrement ses propriétés, font l'hypothèse que le RPN est *sans boucle*, c'est-à-dire qu'aucune place n'est à la fois en pré- et post-condition d'une transition $\forall t \in T, pre(t) \cap post(t) = \emptyset$. Une telle place doit être alors inscrite dans le contexte de la transition. Enfin, pour simplifier les notations, nous faisons l'hypothèse que l'ensemble des places et des transitions sont des entiers de 1 à $|P| + |T|$, autrement dit $P \cup T \equiv \{1, \dots, |P| + |T|\}$.

DÉFINITION.— (Chatain et al. 2019) Étant donné un RPN $N = (P, T, pre, ctx, post, M_0)$ sauf et sans boucle, $\llbracket N \rrbracket$ est le réseau booléen de dimension $n = |P| + |T|$ tel que

$$\begin{aligned} \forall p \in P, \llbracket N \rrbracket_p(x) &= \left(\bigvee_{t \in T: p \in post(t)} x_t \right) \vee \left(x_p \wedge \bigwedge_{t \in T: p \in pre(t)} \neg x_t \right), \\ \forall t \in T, \llbracket N \rrbracket_t(x) &= \left(\bigwedge_{t' \in T} \neg x_{t'} \wedge \bigwedge_{p \in pre(t) \cup ctx(t)} x_p \right) \\ &\quad \vee \left(x_t \wedge \left(\bigvee_{p \in post(t)} \neg x_p \vee \bigvee_{p \in pre(t)} x_p \right) \right). \end{aligned}$$

Étant donné un marquage $M \subseteq P$ de N , la configuration $\llbracket M \rrbracket \in \mathbb{B}^n$ est définie par $\forall p \in M, \llbracket M \rrbracket_p = 1, \forall p \in P \setminus M, \llbracket M \rrbracket_p = 0$, et $\forall t \in T, \llbracket M \rrbracket_t = 0$.

EXEMPLE.— Le réseau de Petri de la figure 1 (p.9) à 4 places et 3 transitions se traduit dans le réseau booléen de dimension 7 suivant :

$$\begin{aligned} f_{p_1}(x) &= x_b \vee (x_{p_1} \wedge \neg x_a) & f_{p_2}(x) &= x_{p_2} \wedge \neg x_c \\ f_{p_3}(x) &= x_a \vee (x_{p_3} \wedge \neg x_b) & f_{p_4}(x) &= x_c \vee x_{p_4} \\ f_a(x) &= (\neg x_a \wedge \neg x_b \wedge \neg x_c \wedge x_{p_1} \wedge x_{p_2}) \vee (x_a \wedge (\neg x_{p_3} \vee x_{p_1})) \\ f_b(x) &= (\neg x_a \wedge \neg x_b \wedge \neg x_c \wedge x_{p_3}) \vee (x_b \wedge (\neg x_{p_1} \vee x_{p_3})) \\ f_c(x) &= (\neg x_a \wedge \neg x_b \wedge \neg x_c \wedge x_{p_2}) \vee (x_c \wedge (\neg x_{p_4} \vee x_{p_2})) \end{aligned}$$

Son marquage initial $(\{p_1, p_2\})$ se traduit par la configuration 1100000.

Le théorème ci-après établit la correspondance entre les marquages et configurations accessibles entre les réseaux de Petri et leur encodage en réseau

booléen, en montrant que la sémantique atomique des réseaux de Petri est équivalente à la sémantique pleinement asynchrone des réseaux booléens, la sémantique par pas des réseaux de Petri est équivalente à la sémantique asynchrone des réseaux booléens, et la sémantique par pas maximal des réseaux de Petri est équivalente à la sémantique synchrone des réseaux booléens. Le fait que les RPN soient sans boucle est crucial pour ces deux dernières correspondances pour assurer qu'avec tout marquage accessible M , tout ensemble de transitions $S \subseteq T$ tel que $\forall t \in S, \llbracket N \rrbracket_t(\llbracket M \rrbracket) = 1$ est un pas admissible dans M .

THÉORÈME.– (Chatain et al. 2019) *Étant donné un RPN $N = (P, T, pre, ctx, post, M_0)$ sauf et sans boucle,*

$$\begin{aligned} M \in \rho_{\text{atom}}^N(M_0) &\iff \llbracket M \rrbracket \in \rho_{\text{a}}^{\llbracket N \rrbracket}(\llbracket M_0 \rrbracket) , \\ M \in \rho_{\text{step}}^N(M_0) &\iff \llbracket M \rrbracket \in \rho_{\text{a}^*}^{\llbracket N \rrbracket}(\llbracket M_0 \rrbracket) , \\ M \in \rho_{\text{mstep}}^N(M_0) &\iff \llbracket M \rrbracket \in \rho_{\text{s}}^{\llbracket N \rrbracket}(\llbracket M_0 \rrbracket) . \end{aligned}$$

0.3.2.3. Discussion

Les réseaux de Petri permettent de spécifier explicitement la causalité des transitions (les conditions pour qu'elles soient possibles). On peut alors mettre en évidence des ensembles de transitions qui peuvent être appliquées dans n'importe quel ordre pour finir toujours sur le même marquage. On parle de *concurrency* : deux transitions distinctes t et t' sont concurrentes si, depuis tout marquage dans lesquels elles sont toutes deux applicables, il est possible à la fois de faire t puis t' , et t' puis t , en atteignant le même nouveau marquage. Les méthodes de *dépliage* (*unfoldings*) prennent avantage de ces transitions concurrentes pour calculer une représentation compacte de l'ensemble des exécutions possibles du réseau (Esparza and Heljanko 2008, Baldan et al. 2012). Ce type d'approche a été appliqué pour analyser les comportements de grands modèles de réseaux biologiques présentant de la concurrence (Chatain et al. 2014, Chatain and Paulevé 2017, Kolčák et al. 2018).

La structure des modèles en réseaux de Petri est très proche des produits synchrones de machines à état finis (automates) (Murata 1989), où plusieurs places mutuellement exclusives (au plus une peut être marquée à la fois) représentent les différents états d'un même automate. Un réseau de Petri n'indique pas explicitement ces différents automates, mais ils peuvent être déduits par un calcul d'invariants de places (par ex., Paulevé 2018).

Enfin, nous attirons l'attention sur l'existence de sémantiques de RPN différentes des sémantiques classiques atomiques et par pas. En particulier la sémantique par intervalle vise à reproduire des comportements existant dans des systèmes où les transitions se déroulent sur des intervalles de temps (Chatain et al. 2015). Cette sémantique peut être définie directement sur les réseaux booléens (Chatain et al.

2018) et montre que les sémantiques usuelles ne capturent pas certains comportements, pourtant potentiellement pertinents.

0.4. Garanties vis-à-vis de modèles quantitatifs

Lorsqu'ils sont appliqués pour la modélisation des systèmes biologiques, les réseaux booléens imposent une forte simplification de l'activité des composants. De nombreux autres formalismes permettent d'intégrer des précisions supplémentaires sur les quantités et vitesses des interactions (réseaux multivalués, réseaux de réactions chimiques, équations différentielles ordinaires, modèles stochastiques, *etc.*). Cependant, leurs spécifications demandent de nombreux paramètres, souvent continus, qui sont généralement inconnus.

L'activité des gènes et la concentration des ARN et protéines n'étant pas binaires, nous pouvons nous demander à quel point une simulation ou une prédiction effectuée à partir d'un réseau booléen peut être transposable au système modélisé, ou à des modèles plus précis.

Idéalement, un modèle abstrait se dérive mathématiquement depuis un modèle initial détaillé (approche *bottom-up*). Des garanties peuvent être dérivées formellement à partir des propriétés des abstractions utilisées (p. ex., Glass and Kauffman 1973, Snoussi 1989, Fages and Soliman 2008, Feret *et al.* 2009, Abou-Jaoudé *et al.* 2016). Cependant, ce schéma ne correspond pas toujours à la façon dont sont employés les réseaux booléens en biologie des systèmes. En effet, en pratique, les réseaux booléens sont souvent construits directement, soit complètement manuellement (à partir de la littérature), soit automatiquement ou semi-automatiquement à partir de bases de données référençant des influences connues entre gènes et protéines, et des données expérimentales. De plus, les réseaux booléens intègrent parfois des processus biologiques abstraits (p. ex., phénotypes) dont l'interaction avec les autres gènes/protéines du réseau serait très difficile à expliciter.

Dans cette section, nous étudions les garanties que peuvent offrir les réseaux booléens vis-à-vis de modèles respectant leur logique tout en rajoutant des informations sur les vitesses et durées des réactions ou sur les quantités des molécules. Nous considérons par la suite deux cadres de modélisation permettant une spécification plus fine de la dynamique d'un ensemble de composants en interaction : les *réseaux multivalués* (Thomas and d'Ari 1990), qui restent purement discrets et qui peuvent être vus comme une généralisation des réseaux booléens, où les composants peuvent prendre une valeur parmi un ensemble borné $\{0, \dots, m\}$; et les *équations différentielles ordinaires* (EDO ; *Ordinary Differential Equations, ODEs*), qui spécifient précisément les évolutions des concentrations des composants au cours du temps (pour les réseaux génétiques, de Jong 2002).

0.4.1. Raffinements de réseaux booléens

Nous caractérisons ici la notion de raffinement d'un réseau booléen f par un réseau multivalué ou par un système d'EDO. Le principe adopté est le suivant : la valeur d'un composant ne peut décroître (resp. croître) dans un réseau multivalué (ou EDO) seulement si le composant peut être mis à 0 (resp. à 1) dans le réseau booléen, depuis une configuration binaire en lien avec la configuration du modèle non booléen.

Introduisons tout d'abord les réseaux multivalués.

DÉFINITION.— Un *réseau multivalué* F de dimension n est une fonction discrète qui associe à toute configuration du réseau l'évolution (unitaire) de la valeur de chaque composant³ :

$$F : \mathbb{M}^n \rightarrow \{-1, 0, 1\}^n \quad \text{où } \mathbb{M} \triangleq \{0, 1, \dots, m\} .$$

De manière similaire aux réseaux booléens, la sémantique synchrone, pleinement asynchrone et asynchrone se définissent comme des relations binaires entre les configurations multivaluées : pour toutes paires de configurations $x, y \in \mathbb{M}^n$

$$x \xrightarrow[s]{F} y \xleftrightarrow{\Delta} x \neq y \wedge \forall i \in [n], y_i = \max(0, \min(m, x_i + F_i(x))) , \quad [5]$$

$$x \xrightarrow[a]{F} y \xleftrightarrow{\Delta} \exists i \in [n] : \Delta(x, y) = \{i\} \wedge y_i = x_i + F_i(x) , \quad [6]$$

$$x \xrightarrow[a^*]{F} y \xleftrightarrow{\Delta} \forall i \in \Delta(x, y), y_i = x_i + F_i(x) . \quad [7]$$

NOTE.— Un réseau booléen f peut être considéré comme un réseau multivalué avec $m = 1$ et F définie telle que pour tout $i \in [n]$, $F_i : x \mapsto \begin{cases} -1 & \text{si } \neg f_i(x) \\ 1 & \text{si } f_i(x) \end{cases}$.

Le critère de raffinement repose sur une *binarisation* des configurations multivaluées. La binarisation associe nécessairement la valeur 0 au booléen 0 et m au booléen 1, et est libre pour les valeurs intermédiaires. On note $\beta : \mathbb{M}^n \rightarrow 2^{\mathbb{B}^n}$ l'ensemble des configurations binaires associées à une configuration multivaluée $x \in \mathbb{M}^n$:

$$\beta(x) \triangleq \{x' \in \mathbb{B}^n \mid \forall i \in [n], x_i = 0 \Rightarrow x'_i = 0 \wedge x_i = m \Rightarrow x'_i = 1\} . \quad [8]$$

3. Pour simplifier les notations, mais sans perte de généralité, la valeur maximale m est fixe pour tous les composants ; en pratique elle peut être différente.

DÉFINITION.— Un réseau multivalué F est un *raffinement* d'un réseau booléen f de même dimension n si et seulement si, pour chaque configuration $x \in \mathbb{M}^n$, et chaque composant $i \in [n]$:

$$\begin{aligned} F_i(x) < 0 &\implies \exists x' \in \beta(x) : \neg f_i(x') \\ \wedge F_i(x) > 0 &\implies \exists x' \in \beta(x) : f_i(x') . \end{aligned}$$

Comme elle repose sur le signe de la dérivée, cette caractérisation de raffinement des réseaux booléens peut être directement étendue aux systèmes d'EDO : de manière analogue aux réseaux multivalués, les EDO spécifient la dérivée de la valeur réelle de chaque composant au cours du temps continu t :

$$\frac{d\mathbb{F}(t, x)}{dt} = \mathcal{F}(x) \quad \text{avec } \mathcal{F} : \mathbb{R}_{\geq 0}^n \rightarrow \mathbb{R}^n . \quad [9]$$

Ici, $\mathcal{F}(x)$ est la dérivée de $\mathbb{F}(t, x)$ selon le temps t en fonction des configurations continues x ; \mathbb{F} étant généralement inconnue.

Seule la binarisation β doit être légèrement adaptée pour refléter l'absence *a priori* de borne maximal ($\beta(x) \triangleq \{x' \in \mathbb{B}^n \mid \forall i \in [n], x_i = 0 \Rightarrow x'_i = 0\}$). La définition de raffinement est alors identique à celle ci-dessus. On peut enfin noter que la sémantique des EDO peut être vue comme analogue à la sémantique *synchrone* : \mathcal{F} décrit l'évolution simultanée de tous les composants.

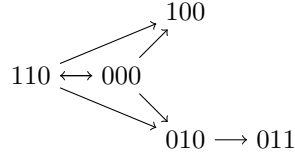
0.4.2. Contre-exemple pour les sémantiques classiques

Comme démontré par le petit exemple qui suit, il s'avère que les réseaux booléens n'offrent aucune garantie sur les comportements prédits avec la sémantique asynchrone (qui englobe les comportements synchrones et pleinement asynchrones). En effet, le raffinement d'un réseau booléen peut à la fois supprimer des comportements possibles, mais également en créer de nouveaux : la sémantique asynchrone booléenne peut manquer certains comportements en fait possibles. Ainsi toute conclusion sur l'existence ou l'absence de trajectoires asynchrones au niveau du réseau booléen ne peut pas toujours se transposer aux modèles raffinés : la sémantique asynchrone ne constitue ni une approximation supérieure (*over-approximation*), ni une approximation inférieure (*under-approximation*) des comportements des modèles raffinés.

Considérons le réseau booléen f de dimension 3 suivant :

$$f_1(x) \triangleq \neg x_2 \quad f_2(x) \triangleq \neg x_1 \quad f_3(x) \triangleq \neg x_1 \wedge x_2$$

Partant de la configuration 000, l'ensemble complet de la relation binaire \xrightarrow{f}_{a^*} de la sémantique asynchrone est le suivant :



Ainsi, $\rho_{a^*}^f(000) = \{000, 110, 010, 011, 100\}$.

Le réseau multivalué F à 3 valeurs ($m = 2$) ci-après est un raffinement de f :

$$F_1(x) \triangleq 1 \text{ si } x_2 < 2 \text{ sinon } -1$$

$$F_2(x) \triangleq 1 \text{ si } x_1 < 2 \text{ sinon } -1$$

$$F_3(x) \triangleq 1 \text{ si } x_1 < 2 \wedge x_2 \geq 1 \text{ sinon } -1$$

Avec la sémantique pleinement asynchrone, l'exécution suivante est possible depuis la configuration 000 :

$$000 \xrightarrow{a}^F 010 \xrightarrow{a}^F 110 \xrightarrow{a}^F 111 \xrightarrow{a}^F 112 \dots$$

Il est ainsi possible d'atteindre une configuration où les 3 composants ne sont pas inactifs, ce qui n'a pas été prédit au niveau booléen ! Dans le cas où la validation de f dépendait de cette possibilité, f aurait été considéré comme invalide, à tord.

NOTE.— Avec la sémantique asynchrone, il est même possible ici d'atteindre la configuration 222 :

$$000 \xrightarrow{a^*}^F 010 \xrightarrow{a^*}^F 110 \xrightarrow{a^*}^F 111 \xrightarrow{a^*}^F 222 \dots$$

Imaginons maintenant un quatrième composant avec $f_4(x) \triangleq x_1 \wedge x_2 \wedge x_3$ et $F_4(x) \triangleq 1$ si $x_1 \geq 1 \wedge x_2 \geq 1 \wedge x_3 \geq 1$ sinon -1 . Dans le réseau multivalué F la sémantique pleinement asynchrone prédit qu'il est possible d'activer ce composant ; contrairement à l'analyse booléenne qui prédit qu'il est impossible de l'activer. Ainsi, sans connaissances suffisantes pour définir directement un réseau multivalué, le modélisateur aurait rejeté le réseau booléen f si les données expérimentales montraient une activation de ce quatrième composant, alors qu'il constitue un modèle potentiellement pertinent du système biologique.

Ce contre-exemple montre une limitation des réseaux booléens avec les sémantiques classiques : elles ne permettent pas d'abstraire correctement la notion de seuils d'influences ni d'ordre entre seuils. Dans l'exemple, le modèle multivalué spécifie que le composant 2 agit à des seuils différents sur 1 et 3 : il suffit que 2 soit dans l'état 1 (ou 2) pour contribuer positivement à 3, mais il doit être nécessairement dans l'état 2 pour contribuer négativement à 1 ($F_1(x) = 1$ si $\neg(x_2 \geq 2)$ sinon -1). Ainsi, lorsque le composant 2 est dans l'état 1, les composants 1 et 3 peuvent augmenter : il est suffisamment actif pour 3 mais pas encore pour 1. Cette ambivalence est impossible à reproduire au niveau purement booléen, où soit 2 est complètement inactif et 1 augmente et 3 diminue, soit complètement actif et 1 diminue et 3 augmente.

Enfin, il est à noter que le réseau booléen de ce contre-exemple se retrouve inclus dans différents modèles de réseaux biologiques publiés (p.ex. Mai and Liu 2009, Martínez-Sosa and Mendoza 2013, Traynard *et al.* 2016). Un problème similaire apparaît avec la modélisation booléenne de boucles d'anticipation incohérentes (Paulevé *et al.* 2020).

0.4.3. Réseaux booléens MP

Avec les sémantiques de mise à jour, un composant booléen va basculer directement depuis son état minimal à son état maximal ou inversement. Comme illustré dans la section précédente, ceci masque certains comportements possibles dans des états intermédiaires, qui peuvent être capturés dans des raffinements multivalués.

La sémantique *la plus permissive* (MP, *most permissive semantics*; Paulevé *et al.* (2020)) est une nouvelle interprétation des réseaux booléens qui permet de capturer ces comportements, sans ajouter d'information ou de paramètres au réseau booléen. En particulier cette sémantique garantit de capturer tout comportement réalisable par tout raffinement quantitatif du réseau booléen (au sens de la Section 0.4.1) avec toute sémantique de mise à jour.

Autrement dit, un raffinement du réseau booléen ne peut que supprimer des comportements observables dans cette abstraction booléenne, mais ne peut pas créer de nouveaux comportements. Les réseaux booléens avec la sémantique MP (raccourcis en réseaux booléens MP; *most permissive Boolean networks*) apportent ainsi une approximation supérieure des comportements. En conséquence, si l'analyse avec la sémantique MP conclut qu'aucune trajectoire n'existe entre deux configurations données, alors aucun raffinement ne peut rendre possible cette trajectoire. Il s'agit actuellement de la seule sémantique offrant ce niveau de garantie.

La sémantique MP repose sur deux notions clés : le changement d'état des composants est décomposé en deux temps, avec un état intermédiaire indiquant une

augmentation \nearrow ou une diminution \searrow . L'ensemble des états possibles d'un composant est noté $\mathbb{P} \triangleq \{0, \nearrow, \searrow, 1\}$. Lorsqu'un composant est dans un état intermédiaire, les autres composants du réseau peuvent lire arbitrairement 0 ou 1. Par exemple, si un composant i est dans l'état intermédiaire \nearrow , un composant j peut l'évaluer à 1 alors qu'un composant k peut l'évaluer à 0. Ceci permet de considérer implicitement le fait que i peut avoir différents seuils d'influence selon les composants du réseau.

La fonction $\gamma : \mathbb{P}^n \rightarrow 2^{\mathbb{B}^n}$ associe à toute configuration $x \in \mathbb{P}^n$ l'ensemble des configurations booléennes qui lui correspondent :

$$\gamma(x) \triangleq \{x' \in \mathbb{B}^n \mid \forall i \in [n], x_i \in \mathbb{B} \Rightarrow x'_i = x_i\} . \quad [10]$$

Par exemple, $\gamma(1 \nearrow 0) = \{100, 110\}$ et $\gamma(\nearrow \searrow \nearrow) = \mathbb{B}^3$.

Nous définissons d'abord cette sémantique en termes de relation binaire entre configurations dans \mathbb{P}^n ; puis nous donnerons une définition alternative à l'aide d'hypercubes clos par f (*trap spaces*).

DÉFINITION.— *Sémantique MP de f*

$$\forall x, y \in \mathbb{P}^n, \quad x \xrightarrow[\text{mp}]{f} y \iff \exists i \in [n] : \Delta(x, y) = \{i\} \wedge y_i = \begin{cases} \nearrow & \text{si } x_i \neq 1 \wedge \exists x' \in \gamma(x) : f_i(x') \\ 1 & \text{si } x_i = \nearrow \\ \searrow & \text{si } x_i \neq 0 \wedge \exists x' \in \gamma(x) : \neg f_i(x') \\ 0 & \text{si } x_i = \searrow . \end{cases} \quad [11]$$

NOTE.— La sémantique telle que définie ci-dessus effectue des changements d'état de façon pleinement asynchrone, mais c'est équivalent de la définir en asynchrone : on remplace la condition $\exists i \in [n] : \Delta(x, y) = \{i\} \wedge y_i = \dots$ par $x \neq y \wedge \forall i \in [n] : y_i = \dots$.

En résumé, un composant $i \in [n]$ dans l'état booléen 0 peut passer dans l'état \nearrow s'il existe une façon x' d'évaluer la configuration (parmi $\gamma(x)$) telle que $f_i(x') = 1$; depuis \nearrow , il peut soit passer dans l'état booléen 1 sans condition, soit dans l'état \searrow s'il existe une façon x' d'évaluer la configuration telle que $f_i(x') = 0$; depuis l'état booléen 1, il peut passer dans l'état \searrow selon cette même dernière condition; enfin depuis l'état \searrow , il peut soit passer dans l'état booléen 0 sans condition, soit dans l'état \nearrow avec les mêmes conditions que le passage de 0 vers \nearrow . La figure 3 résume l'automate correspondant pour chaque composant.

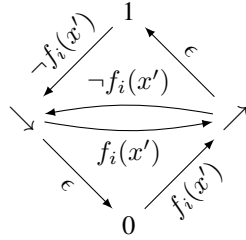


Figure 3. Automate des changements d'état d'un composant i du réseau f avec la sémantique la MP dans une configuration x . Les étiquettes sur les arcs indiquent la condition nécessaire pour le changement d'état, où x' est une configuration appartenant à $\gamma(x)$ et ϵ indique que le changement peut s'effectuer sans condition.

Appliquée au réseau booléen f de la section précédente, la sémantique MP produit la séquence de configurations suivante :

$$000 \xrightarrow[\text{mp}]{} \nearrow 00 \xrightarrow[\text{mp}]{} \nearrow \searrow 0 \xrightarrow[\text{mp}]{} \nearrow 10 \xrightarrow[\text{mp}]{} \nearrow 1 \nearrow \xrightarrow[\text{mp}]{} \nearrow 11 \xrightarrow[\text{mp}]{} \nearrow 111 \dots$$

La configuration 111 est bien prédite, contrairement aux autres sémantiques. Dans la configuration $\nearrow 10$, le composant 3 peut lire la valeur 0 pour le composant 1, ce qui lui permet de passer dans l'état \nearrow et d'atteindre, transitoirement, la configuration 111.

Une abstraction correcte des raffinements multivalués

Comme annoncé en début de section, la sémantique MP garantit de capturer tout comportement possible avec n'importe quel raffinement de f . Pour un raffinement multivalué $F : \mathbb{M}^n \rightarrow \{-1, 0, 1\}^n$ fixé, la fonction $\alpha : \mathbb{M}^n \rightarrow 2^{\mathbb{P}^n}$ associe à toute configuration $x \in \mathbb{M}^n$ l'ensemble des configurations de la sémantique MP telles que : si un composant a une valeur extrême (0 ou m), il a la valeur booléenne correspondante, sinon les deux états intermédiaires \nearrow et \searrow sont considérés :

$$\alpha(x) \triangleq \{\hat{x} \in \mathbb{P}^n \mid \forall i \in [n], x_i = 0 \Leftrightarrow \hat{x}_i = 0 \wedge x_i = m \Leftrightarrow \hat{x}_i = 1\}$$

Le théorème suivant établit que pour toutes configurations multivaluées $x, y \in \mathbb{M}^n$ telles que $x \xrightarrow[\text{a*}]{} y$, alors quelle que soit la configuration MP \hat{x} choisie parmi $\alpha(x)$, il existe une configuration correspondante \hat{y} dans $\alpha(y)$ telle que $\hat{y} \in \rho_{\text{mp}}^f(\hat{x})$.

THÉORÈME.– (Paulevé et al. 2020) Soit f un réseau booléen de dimension n et $F : \mathbb{M}^n \rightarrow \{-1, 0, 1\}^n$ un raffinement multivalué. Alors,

$$\forall x, y \in \mathbb{M}^n, \quad x \xrightarrow[\text{a*}]{F} y \implies \forall \hat{x} \in \alpha(x), \exists \hat{y} \in \alpha(y) : \hat{x} \xrightarrow[\text{mp}]{f} \hat{y}$$

$$\text{où } \forall i \in [n], \quad \hat{y}_i = \begin{cases} \hat{x}_i & \text{si } y_i = x_i \\ \nearrow & \text{si } y_i > x_i \wedge y_i < m \\ 1 & \text{si } y_i > x_i \wedge y_i = m \\ \searrow & \text{si } y_i < x_i \wedge y_i > 0 \\ 0 & \text{si } y_i < x_i \wedge y_i = 0. \end{cases}$$

La démonstration découle de la définition de raffinement : pour chaque composante $i \in [n]$, si $y_i > x_i$ (resp. $y_i < x_i$), forcément $F_i(x) > 0$ (resp. $F_i(x) < 0$). Donc, par la propriété de raffinement, il existe une binarisation $x' \in \beta(x)$ telle que $f_i(x) = 1$ (resp. $f_i(x) = 0$). Il reste à remarquer que pour tout $\hat{x} \in \alpha(x)$, $x' \in \beta(\hat{x})$. Alors, par définition de la sémantique MP, $\hat{x} \xrightarrow[\text{mp}]{f} \hat{y}$.

Il est important de remarquer que le théorème considère que le modèle raffiné peut être mis à jour de manière asynchrone, et donc englobe toute restriction (synchrone, pleinement asynchrone, séquentiel, *etc.*). Sous l'hypothèse que le modèle raffiné s'interprète uniquement selon la mise à jour pleinement asynchrone, une restriction de la sémantique MP est suffisante pour capturer leurs comportements (Chatain *et al.* 2019).

La preuve reposant uniquement sur le signe de la dérivée donnée par le raffinement de f , elle s'étend naturellement aux raffinements par EDO, qui peuvent être vus ici comme un cas limite des réseaux multivalués, appliqués avec la sémantique synchrone.

CORROLAIRE.– Soit f un réseau booléen de dimension n et $\mathcal{F} : \mathbb{R}_{\geq 0}^n \rightarrow \mathbb{R}^n$ un système d'EDO qui raffine f . Alors,

$$\forall x \in \mathbb{R}_{\geq 0}^n, \quad \forall \hat{x} \in \alpha(x), \hat{x} \xrightarrow[\text{mp}]{f} \hat{y}$$

$$\text{où } \forall i \in [n], \hat{y}_i = \begin{cases} \nearrow & \text{si } F_i(x) > 0 \\ \searrow & \text{si } F_i(x) < 0 \wedge x_i > 0 \\ \hat{x}_i & \text{sinon} \end{cases}$$

$$\text{et } \alpha(x) \triangleq \{\hat{x} \in \mathbb{P}^n \mid \forall i \in [n], x_i = 0 \Leftrightarrow \hat{x}_i = 0 \wedge \hat{x}_i \neq 1\}.$$

Définition équivalente sans états intermédiaires

L'ajout des états intermédiaires \nearrow et \searrow pourrait suggérer que la sémantique MP est proche d'un réseau multivalué avec 4 valeurs. Mais ce n'est pas le cas : les états \mathbb{P} ne sont pas totalement ordonnés par les transitions autorisées par la sémantique, comme c'est le cas des réseaux multivalués où les changements d'états sont unitaires.

La sémantique MP peut également se définir en termes d'hypercubes clos par f . Un hypercube est une partie de \mathbb{B}^n où un ensemble de composantes ont une valeur constante, et les autres sont libres (notées $*$) :

DÉFINITION.— Un hypercube h de dimension n est un vecteur dans $(\mathbb{B} \cup \{*\})^n$. L'ensemble des configurations associées est dénotée par $c(h) \triangleq \{x \in \mathbb{B}^n \mid \forall i \in [n], h_i \neq * \Rightarrow x_i = h_i\}$.

Étant donnés deux hypercubes $h, h' \in (\mathbb{B} \cup \{*\})^n$, h est *plus petit* que h' si et seulement si $c(h) \subseteq c(h')$, ou de manière équivalente, $\forall i \in [n], h'_i \neq * \Rightarrow h_i = h'_i$. Un hypercube est *minimal* s'il n'existe pas d'hypercube différent plus petit que lui.

Un hypercube $h \in (\mathbb{B} \cup \{*\})^n$ est *clos* par $f : \mathbb{B}^n \rightarrow \mathbb{B}^n$ si pour toute configuration $x \in c(h)$, $f(x) \in c(h)$. Nous généralisons cette dernière définition pour n'imposer la clôture que pour un sous-ensemble de composantes $K \subseteq [n]$:

DÉFINITION.— Étant donné $K \subseteq [n]$, un hypercube $h \in (\mathbb{B} \cup \{*\})^n$ est *K-clos* par f si pour toute configuration $x \in c(h)$, pour toute composante $i \in K$, $h_i \in \{*, f_i(x)\}$.

NOTE.— Un hypercube h est clos par f si et seulement si h est $[n]$ -clos par f . Un hypercube h clos par f est minimal si et seulement si c'est le plus petit hypercube clos par f contenant x , pour tout $x \in c(h)$.

EXEMPLE.— Soit $f : \mathbb{B}^3 \rightarrow \mathbb{B}^3$ telle que $f_1(x) \triangleq \neg x_2$, $f_2(x) \triangleq \neg x_1$, et $f_3(x) \triangleq \neg x_1 \wedge x_2$. L'hypercube $01*$ est clos par f , avec $c(01*) = \{010, 011\}$. L'hypercube $1*0$ est le plus petit hypercube $\{2, 3\}$ -clos par f contenant 110 ; il n'est pas clos par f , ni le plus petit $\{2, 3\}$ -clos par f contenant 100 .

Partant d'une configuration x , la sémantique MP peut s'exprimer avec le calcul des plus petits hypercubes contenant x et K -clos par f , pour tout ensemble K :

- x est l'unique hypercube \emptyset -clos par f contenant x ;
- le changement d'état d'un composant $i \in [n]$ vers \nearrow ou \searrow produit une configuration x' où $\gamma(x')$ correspond à l'hypercube $h \in (\mathbb{B} \cup \{*\})^n$ avec $h_i = *$ et pour tout autre composante $j \in [n], j \neq i$, $h_j = x_j$. Ainsi, h est le plus petit hypercube $\{i\}$ -clos par f et contenant x .
- en considérant seulement les changements d'états vers \nearrow ou \searrow , la sémantique va progressivement agrandir l'hypercube selon les composantes modifiées, et ainsi construire les plus petits hypercubes K -clos par f et contenant x , pour tout $K \subseteq [n]$.

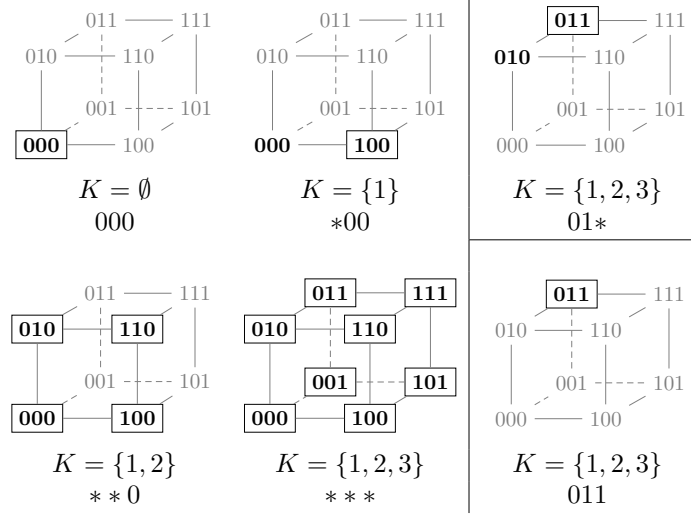


Figure 4. Exemples de plus petits hypercubes K -clos contenant la configuration 000 (gauche), 010 (haut droite) et 011 (bas droite) pour le réseau booléen f de dimension 3 avec $f_1(x) \triangleq \neg x_2$, $f_2(x) \triangleq \neg x_1$, $f_3(x) \triangleq \neg x_1 \wedge x_2$. Les configurations appartenant à l'hypercube sont indiquées en gras ; celles vérifiant la propriété d'accessibilité sont encadrées. Seul l'hypercube 011 est clos par f et minimal.

Avec la sémantique MP, le changement d'état d'un composant depuis \nearrow ou \searrow vers un état booléen est sans condition et est complètement déterminé par l'état intermédiaire : 1 depuis \nearrow et 0 depuis \searrow . Or, partant d'une configuration initiale booléenne, un composant peut être dans l'état \nearrow seulement si une configuration précédente $x' \in \mathbb{P}^n$ était telle qu'un $z \in \gamma(x')$ vérifiait $f_i(z) = 1$ (resp. \searrow si $f_i(z) = 0$).

Ainsi, une configuration $y \in \mathbb{B}^n$ est atteinte depuis $x \in \mathbb{B}^n$ avec la sémantique MP si et seulement s'il existe un ensemble de composantes $K \subseteq [n]$ tel que le plus petit hypercube $h \in (\mathbb{B} \cup \{*\})^n$ contenant x et K -clos par f vérifie les deux conditions suivantes : (1) il contient y , et (2) pour toute composante $i \in K$, il existe une configuration z dans h ($z \in c(h)$) telle que $f_i(z) = y_i$.

La figure 4 illustre ce calcul d'hypercubes avec le réseau booléen f défini dans la Section 0.4.2.

Pour conclure, nous pouvons nous interroger sur la pertinence de la sémantique MP : une sémantique qui rend toute configuration accessible depuis toute autre configuration aurait les mêmes garanties vis-à-vis de raffinements, mais serait inutile

à analyser. Dans la section suivante, nous apportons quelques éléments de réponses montrant que la sémantique MP reste tout de même restrictive et préserve des aspects importants de la dynamique des systèmes biologiques, en particulier pour les phénomènes de différenciation.

0.5. Propriétés dynamiques et complexités

L'analyse de la dynamique des réseaux booléens repose le plus souvent sur trois propriétés fondamentales : les points fixes, caractérisant les configurations qui ne peuvent plus évoluer ; l'accessibilité, caractérisant l'existence de trajectoires entre deux configurations données ; et les attracteurs, généralisant les points fixes, et caractérisant les comportements limites du réseau.

Cette section liste quelques résultats de complexité en lien avec ces propriétés : concernant les sémantiques classiques, synchrone, pleinement asynchrone et asynchrone, le problème d'existence de point fixe est NP-complet, et le problème de décision des propriétés d'accessibilité et d'appartenance à un attracteur sont PSPACE-complets.

Concernant la sémantique MP, le problème d'existence de points fixes est NP-complet ; le problème de décision de propriétés d'accessibilité est P^{NP} dans le cas général et P dans le cas des réseaux booléens localement monotones ; enfin le problème de décision de l'appartenance d'une configuration à un attracteur est $coNP^{coNP}$ dans le cas général et coNP dans le cas des réseaux booléens localement monotones.

0.5.1. Points fixes

Les points fixes des réseaux booléens sont une des propriétés les plus systématiquement étudiée. Pour la modélisation en biologie, ces points fixes correspondent à des régimes particulièrement stables de la cellule, où tous les composants restent qualitativement constants.

DÉFINITION.— Une configuration $x \in \mathbb{B}^n$ est un *point fixe* du réseau booléen f avec la sémantique σ si $\rho_\sigma^f(x) = \{x\}$.

Pour les sémantiques de mise à jour synchrone, pleinement asynchrone, et asynchrone, ainsi que pour la sémantique MP, leurs points fixes sont identiques aux points fixes de f , c'est-à-dire aux configurations $x \in \mathbb{B}^n$ telles que $f(x) = x$.

THÉORÈME.— Déterminer si le réseau booléen f possède au moins un point fixe est un problème NP-complet avec la sémantique synchrone, pleinement asynchrone, asynchrone, et MP.

La preuve peut s'effectuer par réduction du problème SAT (Formenti *et al.* 2014).

En pratique, le calcul de points fixes de la fonction f peut s'effectuer par exemple à l'aide de manipulations de diagrammes de décision binaire (Naldi *et al.* 2007) ou encore par résolution de problème de satisfaction booléenne (SAT) (Chevalier *et al.* 2019), et est applicable à des réseaux de très grande dimension ($n > 1000$).

0.5.2. Accessibilité entre configurations

L'existence d'au moins une trajectoire menant d'une configuration $x \in \mathbb{B}^n$ à une autre configuration $y \in \mathbb{B}^n$ est appelé accessibilité (*reachability*) :

DÉFINITION.— Étant données deux configurations $x, y \in \mathbb{B}^n$ et une sémantique σ , y est *accessible* depuis x si $y \in \rho_\sigma^f(x)$.

Pour les applications en biologie, deux familles de propriétés liées à l'accessibilité sont couramment employées : les observations de marqueurs au sein d'une cellule au cours du temps (séries temporelles) se traduisent en propriétés d'accessibilité *positives* (existence) ; alors que les observations de phénomènes de différenciation se traduisent en propriétés d'accessibilité *négatives* (absence) : sans perturbation externe, une cellule différenciée dans un sous-type A ne peut pas atteindre un état correspondant au sous-type B, ni un état pluripotent précédent.

Sémantiques de mise à jour

Avec les sémantiques classiques, il s'avère que le problème d'accessibilité est PSPACE-complet, et donc extrêmement difficile à résoudre pour des réseaux de grande dimension, même comparé à l'existence de points fixes⁴.

THÉORÈME.— Déterminer si $y \in \rho_\sigma^f(x)$ est un problème PSPACE-complet avec les sémantiques de mise à jour synchrone, pleinement asynchrone et asynchrone ($\sigma \in \{s, a, a^*\}$).

Intuitivement, la complexité peut se comprendre par le fait que deux configurations peuvent être liées seulement par des trajectoires de taille exponentielle avec n (Melliti *et al.* 2016). Ainsi, même avec la sémantique synchrone qui est déterministe, une simulation peut prendre un temps exponentiel pour la vérification.

Le nombre de configurations accessibles étant au plus 2^n , ce problème est au plus PSPACE⁵ car il suffit de simuler au plus $2^n - 1$ itérations du réseau booléen de manière non déterministe avec un compteur sur n bits.

4. Il n'est pas (encore) connu si NP=PSPACE ; mais en pratique la résolution de problèmes NP (SAT) peut s'appliquer à des problèmes avec des millions de variables, alors que la résolution de problèmes PSPACE (p. ex., model-checking) se limite à quelques centaines de variables.

5. PSPACE=NSPACE=coNSPACE (Papadimitriou 1995).

Plusieurs preuves sur des modèles similaires aux réseaux booléens permettent de conclure que le problème d'accessibilité est PSPACE-complet. Dans le cas synchrone, cela découle directement du problème d'accessibilité dans les systèmes de réactions, une sous-classe des réseaux booléens synchrones, démontré PSPACE-complet par Dennunzio *et al.* (2019).

Pour le cas asynchrone et pleinement asynchrone, outre mentionner l'analogie avec le problème d'accessibilité dans les automates cellulaires finis asynchrones et pleinement asynchrones, démontré PSPACE-complet par Dennunzio *et al.* (2015), l'aspect PSPACE-difficile peut se démontrer, entre autres, de plusieurs façons :

- Par réduction du problème d'accessibilité dans les réseaux de Petri saufs sans arcs de lecture, démontré PSPACE-complet par Cheng *et al.* (1995) avec la sémantique atomique (et donc également par pas, d'après la remarque en fin d'introduction de la Section 0.3.2), avec l'encodage donné dans la Section 0.3.2.2 (Chatain *et al.* 2019).

- Par réduction du problème d'accessibilité avec la sémantique synchrone. Tout comme pour les automates cellulaires (Nakamura 1981), il est possible de construire un réseau booléen f' dont les sémantiques asynchrone et pleinement asynchrone préservent les propriétés d'accessibilité d'un réseau booléen f avec la sémantique synchrone, avec un nombre de dimensions polynomial avec n .

Le principe général d'une construction possible est de décomposer une itération synchrone en plusieurs étapes qui peuvent être effectuées de manière asynchrone. Trois étapes sont à discerner : (a) le calcul de la valeur suivante pour chaque composante $i \in [n]$; (b) l'application du nouvel état pour chaque composant; (c) la remise à zéro des composantes supplémentaires employés pour la décomposition. Cela peut s'effectuer avec un réseau f' avec $3n + 2$ dimensions : une composante z dont l'état 1 signale la remise à zéro des composantes supplémentaires (hormis z); une composante w dont l'état 0, sous condition que z est dans l'état 0, signale l'étape de calcul, et l'état 1 l'étape d'application. Pour chaque composante $i \in [n]$ du réseau initial, deux composantes ci et $\bar{c}i$ sont ajoutées, dont l'état 1 indique respectivement si f_i est vrai ou faux. La fin de l'étape de calcul est déterminée par le fait que pour toute composante $i \in [n]$, soit ci soit $\bar{c}i$ est dans l'état 1 ; la composante w passe alors dans l'état 1 ; les composantes i passent alors à l'état 0 si et seulement si $\bar{c}i$ vaut 1 et à l'état 1 si et seulement si ci vaut 1. La fin de l'étape d'application est déterminée par le fait que toutes les composantes i ont la valeur calculée ; la composante z passe alors dans l'état 1 qui va déclencher la transition vers l'état 0 des composants w et ci et $\bar{c}i$. Enfin, la composante z repasse dans l'état 0, ce qui permet aux composantes ci et $\bar{c}i$ de calculer la prochaine valeur de chaque composante $i \in [n]$. Un tel réseau $f' : \mathbb{B}^{3n+2} \rightarrow \mathbb{B}^{3n+2}$ peut se définir de la façon suivante, où $x_{1..n}$ dénote la

configuration x tronquée aux n premières composantes :

$$\begin{aligned}
f'_i(x') &= ((\neg x'_w \vee x'_z) \wedge x'_i) \vee (x'_w \wedge \neg x'_z \wedge x'_{\bar{c}i}) \\
f'_{c_i}(x') &= \neg x'_z \wedge ((\neg x'_w \wedge f_i(x'_{1..n})) \vee (x'_w \wedge x'_{c_i})) \\
f'_{\bar{c}i}(x') &= \neg x'_z \wedge ((\neg x'_w \wedge \neg f_i(x'_{1..n})) \vee (x'_w \wedge x'_{\bar{c}i})) \\
f'_w(x') &= \neg x'_z \wedge \left(\left(x'_w \vee \bigwedge_{i \in [n]} (x'_{c_i} \vee x'_{\bar{c}i}) \right) \right) \\
f'_z(x') &= \left(x'_w \wedge \bigwedge_{i \in [n]} (x'_{c_i} \Leftrightarrow x'_i \wedge x'_{\bar{c}i} \Leftrightarrow \neg x'_i) \right) \\
&\quad \vee \left(x'_z \wedge \left(x'_w \vee \bigvee_{i \in [n]} (x'_{c_i} \vee x'_{\bar{c}i}) \right) \right)
\end{aligned}$$

Les sémantiques asynchrone et pleinement asynchrone du réseau f' préservent exactement les propriétés d'accessibilité de la sémantique synchrone de f :

$$y \in \rho_s^f(x) \iff y0^{2n+2} \in \rho_{a*}^{f'}(x0^{2n+2}) \iff y0^{2n+2} \in \rho_a^{f'}(x0^{2n+2}) \quad [12]$$

pour toutes configurations $x, y \in \mathbb{B}^n$, où 0^{2n+2} est le vecteur 0 de dimension $2n+2$ et $y0^{2n+2}$ dénote sa concaténation à y .

En pratique, la vérification de propriétés d'accessibilité peut s'effectuer par des représentations symboliques de l'ensemble des configurations accessibles, notamment avec des structures de données inspirées des diagrammes de décision binaires (Hamez *et al.* 2009). Une approche complémentaire est l'exploration en ordre partiel des changements d'états possibles, notamment avec les dépliages des réseaux de Petri (Esparza and Heljanko 2008, Haar *et al.* 2013, Rodríguez and Schwoon 2013). Ces méthodes permettent d'analyser des réseaux booléens avec quelques centaines de dimensions, selon la structure du réseau et l'emploi d'analyses statiques pour réduire le modèle (Paulevé 2018, Chatain and Paulevé 2017).

Sémantique MP

Déterminer si $y \in \rho_{\text{mp}}^f(x)$ est un problème bien plus simple. Nous allons l'expliquer avec la formulation de la sémantique avec les hypercubes : $y \in \rho_{\text{mp}}^f(x)$ si et seulement s'il existe un ensemble de composantes $K \subseteq [n]$ tel que le plus petit hypercube K -clos $h \in (\mathbb{B} \cup \{*\})^n$ contenant x vérifie (1) $y \in c(h)$ et (2) pour tout $i \in K$, $\exists z \in c(h) : f_i(z) = y_i$.

Pour un K fixé, le calcul du plus petit hypercube K -clos contenant x et la vérification des deux conditions peut s'effectuer selon l'algorithme suivant :

```

h := x
répéter |K| fois: (* construction *)
  pour chaque i ∈ K tel que h_i ≠ *:
    si ∃z ∈ c(h) tel que f_i(z) ≠ x_i:

```

```

      hi := *
pour chaque i ∈ [n]: (* validation *)
  si hi ≠ * et hi ≠ yi:
    répondre non
  si hi = * et si non (∃z ∈ c(h), fi(z) = yi):
    répondre non
répondre oui

```

Il faut d'abord remarquer qu'il n'est pas nécessaire d'explorer tout ensemble $K \subseteq [n]$ possible. En effet, admettons qu'il existe deux composantes distinctes $j, j' \in K$ vérifiant la condition (1) mais ne vérifiant pas la condition (2) : $j \neq j', h_j = h_{j'} = *$ et $\forall z \in c(h), f_j(z) \neq y_j$ et $f_{j'}(z) \neq y_{j'}$. Il apparaît que j ne vérifiera pas la condition (2) avec $K \setminus \{j'\}$ ni j' avec $K \setminus \{j\}$.

Ainsi, l'exploration débute avec $K = [n]$. Si l'algorithme ci-dessus répond non, on soustrait de K les composantes ne vérifiant pas la condition (2). À tout moment, si la condition (1) n'est pas vérifiée, on peut directement conclure à l'absence d'accessibilité, car elle ne le sera pour aucun sous-ensemble de K . La condition (2) sera toujours vérifiée à un moment, au plus tard avec $K = \emptyset$. Au final, l'algorithme de décision est exécuté au plus n fois.

Maintenant, remarquons que l'algorithme de décision effectue plusieurs tests de la forme " $\exists z \in c(h) : f_i(z) = b$ " avec $b \in \mathbb{B}$. Dans le cas général, c'est exactement le problème SAT. Le problème de décision de l'accessibilité est alors dans la classe de complexité P^{NP} . Dans le cas où f est localement monotone (voir introduction), tester si $\exists z \in c(h) : f_i(z) = b$ avec $b \in \mathbb{B}$ donné est linéaire avec n : il suffit de choisir le z maximal pour f_i si $b = 1$ et minimal si $b = 0$ (en prenant les notations de la définition de localement monotone, si $b = 1$, pour chaque $j \in [n]$ où $h_j = *$, on choisit $z_j = 1$ si $\leq_j^i = \leq$ et $z_j = 0$ sinon, et inversement dans le cas où $b = 0$). Le problème de décision de l'accessibilité est alors dans la classe de complexité P.

THÉORÈME.— (Paulevé et al. 2020) *Déterminer si $y \in \rho_{\text{mp}}^f(x)$ est un problème P si f est localement monotone et P^{NP} sinon.*

0.5.3. Attracteurs

Les attracteurs d'un réseau booléen caractérisent ses comportements limites : ce sont les ensembles de configurations où toutes paires de configurations ont une trajectoire entre elles, et où aucune configuration en dehors de l'attracteur n'est accessible depuis l'attracteur. Les attracteurs généralisent ainsi les points fixes (un point fixe est un attracteur composé d'une seule configuration) et reposent sur la notion d'accessibilité abordée dans la section précédente.

DÉFINITION.– Un ensemble non-vide de configurations $A \subseteq \mathbb{B}^n$ est un *attracteur* du réseau booléen f avec une sémantique σ si $\forall x \in A, \rho_\sigma^f(x) = A$.

Tout comme pour les points fixes, les attracteurs modélisent des comportements stables de la cellule, potentiellement oscillatoires.

Nous nous intéressons à la complexité du problème de décision suivant : étant donnée une configuration $x \in \mathbb{B}^n$, x appartient à un attracteur.

Sémantiques de mise à jour

Avec les sémantiques synchrone, pleinement asynchrone, et asynchrone, décider si une configuration x appartient à un attracteur est PSPACE. Une façon simple de le démontrer est d'étudier le problème complémentaire : x n'appartient pas à un attracteur si et seulement si il existe une configuration y telle que y est accessible depuis x , mais x n'est pas accessible depuis y . D'après la complexité de l'accessibilité, ce problème est dans NPSPACE, et donc l'appartenance à un attracteur dans $\text{coNPSPACE} = \text{PSPACE}$.

THÉORÈME.– Déterminer si une configuration $x \in \mathbb{B}^n$ appartient à un attracteur de f est PSPACE-complet avec la sémantique synchrone, asynchrone, et pleinement asynchrone.

Pour le cas synchrone, ce résultat découle de la complexité de ce problème dans le cas des systèmes de réaction synchrones, cas particuliers des réseaux booléens, démontré PSPACE-complet par Dennunzio *et al.* (2019). Pour les cas asynchrones et pleinement asynchrones, cela peut alors se démontrer par réduction du cas synchrone, avec la construction esquissée dans la section précédente sur la complexité de l'accessibilité.

En pratique, le calcul exact des attracteurs des réseaux booléens passe difficilement à l'échelle. Des méthodes reposant sur des représentations symboliques des états accessibles tels que les dépliages de réseaux de Petri (Chatain *et al.* 2014) ou les diagrammes de décision binaires (Mizera *et al.* 2017, 2019), et sont limitées à des dimensions de l'ordre de 50 à 100 selon les spécificités structurelles du réseau et la sémantique choisie.

Sémantique MP

Les attracteurs avec la sémantique MP sont des objets bien particuliers : ils correspondent exactement aux hypercubes minimaux clos par f , aussi appelés *minimal trap spaces*.

LEMME.– $A \subseteq \mathbb{B}^n$ est un attracteur du réseau booléen f avec la sémantique MP si et seulement s'il existe un hypercube $h \in (\mathbb{B} \cup \{*\})^n$ minimal clos par f tel que $c(h) = A$.

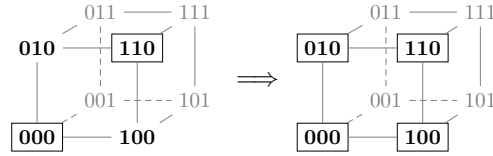


Figure 5. Illustration de la propriété des attracteurs de la sémantique MP, avec la convention graphique de la figure 4 : si deux configurations sur une diagonale d'un hypercube sont accessibles deux à deux et appartiennent à un attracteur, alors les configurations adjacentes appartiennent également à l'attracteur.

Ces hypercubes minimaux ont déjà été appliqués comme approximations des attracteurs de sémantiques de mises à jour (Klärner *et al.* 2015, Naldi 2018). En effet, tout hypercube minimal clos par f contient au moins un attracteur de chaque sémantique de mise à jour ; mais la réciproque n'est pas vraie.

Avec la sémantique MP, les attracteurs deviennent exactement ces hypercubes clos minimaux, ce qui rend leur identification applicable à des très grands réseaux, comme nous allons en discuter ci-après. De plus, on peut remarquer que le nombre d'attracteurs de la sémantique MP est une borne inférieure au nombre d'attracteurs de tout raffinement possible de f .

Une façon de comprendre pourquoi un attracteur de la sémantique MP est forcément un hypercube est de remarquer que si deux configurations sur une diagonale d'un hypercube sont dans un même attracteur (donc accessibles de l'un vers l'autre), alors toutes les configurations sur les sommets adjacents à la diagonale sont dans l'attracteur (figure 5).

Plus précisément, prenons une configuration $x \in \mathbb{B}^n$ dans un attracteur, et soit $h \in (\mathbb{B} \cup \{*\})^n$ le plus petit hypercube clos par f et contenant x . Soit la configuration $y \in c(h)$ la plus distante de x appartenant à l'hypercube : $\forall i \in [n], y_i = \neg x_i$ si $h_i = *$, sinon $y_i = x_i$. D'après la section précédente, y est accessible depuis x ($y \in \rho_{\text{mp}}^f(x)$), et par hypothèse, x est également accessible depuis y . Il reste à remarquer que le plus petit hypercube clos par f et contenant y est alors h . Ainsi, pour toute composante $i \in [n]$ libre dans h ($h_i = *$), il existe une configuration de l'hypercube $z \in c(h)$ telle que $f_i(z) = 0$ et une configuration $z' \in c(h)$ telle que $f_i(z') = 1$. Il en résulte que toute configuration de l'hypercube h est accessible depuis x . Enfin, remarquons que si le plus petit hypercube h clos par f et contenant x n'est pas minimal, alors il existe une configuration $y \in c(h)$ accessible depuis x , mais x n'appartient pas au plus petit hypercube clos par f et contenant y : x n'est donc pas accessible depuis y et n'appartient donc à aucun attracteur.

Le problème de décision d'appartenance d'une configuration x donnée à un attracteur revient alors à décider si le plus petit hypercube clos par f et contenant x est minimal.

Soit N'EST-PAS-CLOS(h) le problème de décider si l'hypercube h n'est pas clos par f , autrement dit si il existe $i \in [n]$ avec $h_i \neq *$ et $z \in c(h)$ tel que $f_i(z) \neq h_i$. Comme nous l'avons vu dans la section précédente, ce problème est NP dans le cas général et P dans le cas où f est localement monotone. Ainsi, le problème complémentaire EST-CLOS(h) est coNP dans le cas général et P dans le cas localement monotone.

Le problème N'EST-PAS-MINIMAL(h) de décider si un hypercube h clos par f est un hypercube minimal peut se résumer à trouver un hypercube h' strictement inclus dans h et clos par f , ce qui est de complexité au plus $\text{NP}^{\text{EST-CLOS}}$. Ainsi, le problème complémentaire EST-MINIMAL(h) est au plus $\text{coNP}^{\text{EST-CLOS}}$: $\text{coNP}^{\text{coNP}} = \Pi_2^P$ dans le cas général et coNP dans le cas localement monotone.

THÉORÈME.— (Paulevé et al. 2020) *Déterminer si une configuration $x \in \mathbb{B}^n$ appartient à un attracteur de f avec la sémantique MP est au plus coNP si f est localement monotone et au plus Π_2^P sinon.*

Le calcul des hypercubes minimaux d'un réseau booléen (et donc des attracteurs de la sémantique MP) peut s'effectuer efficacement en Answer-Set Programming qui intègre la notion de solution minimale par inclusion, et est applicable à des réseaux de très grande échelle (Klarner et al. 2015).

0.6. Conclusion

Ce chapitre a abordé quelques aspects théoriques des réseaux booléens, avec leurs connexions avec des formalismes classiques très étudiés en informatique, leurs liens avec des modèles plus quantitatifs, et la complexité pour déterminer des propriétés liées aux trajectoires et aux comportements limites.

Il apparaît que les sémantiques classiques pour les réseaux booléens produisent des dynamiques à la fois complexes à analyser (la majorité des problèmes étant PSPACE-complets) et sans garanties vis-à-vis de modèles respectant la même logique mais apportant des dimensions quantitatives. La sémantique MP apporte une nouvelle interprétation des réseaux booléens qui résout ces problèmes tout en continuant de capturer des comportements essentiels liés à la multi-stabilité. Elle permet de raisonner au niveau booléen, avec donc très peu de paramètres, et de pouvoir déjà apporter quelques conclusions transposables à tout raffinement possible, sans avoir à les explorer explicitement.

La sémantique MP permet une analyse avec le moins d'hypothèses possibles sur le système sous-jacent, hypothèses souvent utilisées pour des raisons techniques et

sans lien avec les connaissances sur le système biologique modélisé. Il est commun de lire que la sémantique pleinement asynchrone est la plus réaliste biologiquement souvent avec un simple argument d'autorité et l'intuition que cela permet de capturer différentes échelles temporelles dans les influences. Le contre-exemple étudié dans la Section 0.4.2 montre que la mise à jour pleinement asynchrone (tout comme l'asynchrone) peut, de manière contre-intuitive, masquer certains comportements pourtant pertinents biologiquement. La sémantique MP permet de retrouver ces comportements sans avoir à "agrandir" l'espace des états, ce qui nécessiterait en général de déterminer de nombreux paramètres supplémentaires, au prix de potentiels comportements artificiels.

La question de la minimalité de l'abstraction apportée par la sémantique MP, et donc de l'existence de ces comportements artificiels, est intrinsèquement liée aux raffinements admis. Nous avons pu démontrer que dans le cas général *la sémantique MP est minimale* vis-à-vis des propriétés d'accessibilité (Paulevé *et al.* 2020) : si la sémantique MP prédit une accessibilité entre deux configurations x et y , alors il existe un raffinement du réseau booléen qui réalise cette accessibilité avec la sémantique asynchrone. Il serait alors intéressant de considérer différentes classes de raffinements pour lesquels la sémantique MP ajouterait des comportements artificiels.

Les propriétés dynamiques abordées dans ce chapitre sont élémentaires à de nombreuses applications des réseaux booléens en biologie des systèmes, en particulier pour le contrôle et la synthèse.

Le *contrôle* des réseaux booléens consiste à appliquer des perturbations (de configuration ou de fonction) pour garantir l'accessibilité d'un attracteur donné. La prédiction de ces perturbations donne alors des cibles potentielles pour provoquer une dé-différenciation ou trans-différenciation de la cellule (on parle également de *reprogrammation cellulaire*). Le problème du contrôle a de nombreuses variantes, selon sa restriction à partir d'une configuration initiale donnée, ou le type de perturbations (permanentes, temporaires, ou instantanées) et leur application (simultanée, ou en séquence); mais repose toujours fondamentalement sur le calcul d'attracteurs et de propriétés d'accessibilité (Zañudo and Albert 2015, Biane and Delaplace 2018, Mandon *et al.* 2019, Su *et al.* 2019).

La *synthèse* de réseaux booléens consiste à construire automatiquement des réseaux dont la structure et la dynamique vérifient des propriétés fixées *a priori*. Les contraintes sur la structure proviennent généralement de bases de données référençant les interactions potentielles entre les gènes et protéines sélectionnées, ou de méthodes statistiques pour suggérer des interactions possibles à partir d'observations du système. Les contraintes sur la dynamique proviennent le plus souvent de l'interprétation de données biologiques montrant des comportements temporelles (séries temporelles, donnant des propriétés d'accessibilité) et de différenciation (propriétés d'attracteurs et d'absence de trajectoires). Ainsi, à la

complexité d'analyse d'un réseau booléen fixé s'ajoute la combinatoire des réseaux booléens potentiels à explorer. Grâce à la faible complexité de l'analyse de la sémantique MP, nous avons récemment démontré que la synthèse de réseaux booléens peut s'effectuer à très grande échelle pour certaines familles de propriétés dynamiques, tout en ayant des garanties importantes vis-à-vis de modèles issus de raffinements quantitatifs (Chevalier *et al.* 2019).

Le problème de synthèse pose encore de nombreux défis à la fois pour le passage à l'échelle et la prise en compte de propriétés dynamiques complexes, et leur applications à des problématiques pour le contrôle des systèmes biologiques. En pratique, la synthèse aboutit très souvent sur un nombre astronomique de réseaux booléens vérifiant les contraintes imposées (Razzaq *et al.* 2018). Il devient alors crucial de savoir échantillonner efficacement l'espace des solutions pour capturer la diversité des réseaux à considérer, tout en cherchant à assurer une robustesse des prédictions vis-à-vis des données d'apprentissage. Ces défis ouvrent des perspectives excitantes, à la frontière entre méthodes formelles et méthodes statistiques d'apprentissage.

0.7. Remerciements

La plupart des résultats mentionnés sont issus de travaux et discussions avec Stefan Haar et Thomas Chatain (Inria Saclay-Île-de-France, LSV, et École normale supérieure de Paris-Saclay), et plus généralement de collaborations dans le cadre du projet bilatéral France-Luxembourg *AlgoReCell*⁶, co-financé par l'*Agence Nationale pour la Recherche* (ANR ; projet ANR-16-CE12-0034). Je remercie également les relecteurs pour leurs commentaires attentifs.

0.8. Bibliographie

- Abou-Jaoudé, W., Monteiro, P. T., Naldi, A., Grandclaude, M., Soumelis, V., Chaouiya, C., Thieffry, D. (2015), Model checking to assess T-Helper cell plasticity, *Frontiers in Bioengineering and Biotechnology*, 2.
- Abou-Jaoudé, W., Thieffry, D., Feret, J. (2016), Formal derivation of qualitative dynamical models from biochemical networks, *Biosystems*, 149, 70–112.
- Aracena, J., Goles, E., Moreira, A., Salinas, L. (2009), On the robustness of update schedules in Boolean networks, *Biosystems*, 97(1), 1 – 8.
- Baldan, P., Bruni, A., Corradini, A., König, B., Rodríguez, C., Schwoon, S. (2012), Efficient unfolding of contextual Petri nets, *TCS*, 449, 2–22.

6. <http://algorecell.lri.fr>

- Biane, C., Delaplace, F. (2018), Causal reasoning on boolean control networks based on abduction : theory and application to cancer drug discovery, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–1.
- Chaouiya, C., Remy, E., Ruet, P., Thieffry, D. (2004), Qualitative modelling of genetic networks : From logical regulatory graphs to standard petri nets, in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 137–156.
- Chatain, T., Haar, S., Jezequel, L., Paulevé, L., Schwoon, S. (2014), Characterization of reachable attractors using Petri net unfoldings, in P. Mendes, J. Dada, K. Smallbone, (eds), *Computational Methods in Systems Biology*, vol. 8859 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Cham, pp. 129–142.
- Chatain, T., Haar, S., Kolčák, J., Paulevé, L., Thakkar, A. (2019), Concurrency in Boolean networks, *Natural Computing*, .
- Chatain, T., Haar, S., Koutny, M., Schwoon, S. (2015), Non-atomic transition firing in contextual nets, in *Applications and Theory of Petri Nets*, vol. 9115 of *Lecture Notes in Computer Science*, Springer, pp. 117–136.
- Chatain, T., Haar, S., Paulevé, L. (2018), Boolean Networks : Beyond Generalized Asynchronicity, in J. M. Baetens, M. Kutrib, (eds), *Cellular Automata and Discrete Complex Systems (AUTOMATA 2018)*, vol. 10875 of *Lecture Notes in Computer Science*, Springer, Ghent, Belgium, pp. 29–42.
- Chatain, T., Paulevé, L. (2017), Goal-Driven Unfolding of Petri Nets, in R. Meyer, U. Nestmann, (eds), *28th International Conference on Concurrency Theory (CONCUR 2017)*, vol. 85 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 18 :1–18 :16.
- Cheng, A., Esparza, J., Palsberg, J. (1995), Complexity results for 1-safe nets, *Theoretical Computer Science*, 147(1&2), 117–136.
- Chevalier, S., Froidevaux, C., Paulevé, L., Zinovyev, A. (2019), Synthesis of Boolean Networks from Biological Dynamical Constraints using Answer-Set Programming, in *31st International Conference on Tools with Artificial Intelligence, ICTAI, IEEE*, Portland, Oregon, United States.
- Cohen, D. P. A., Martignetti, L., Robine, S., Barillot, E., Zinovyev, A., Calzone, L. (2015), Mathematical modelling of molecular pathways enabling tumour cell invasion and migration, *PLoS Comput Biol*, 11(11), e1004571.
- Collombet, S., van Oevelen, C., Sardina Ortega, J. L., Abou-Jaoudé, W., Di Stefano, B., Thomas-Chollier, M., Graf, T., Thieffry, D. (2017), Logical modeling of lymphoid and myeloid cell specification and transdifferentiation, *Proceedings of the National Academy of Sciences*, 114(23), 5792–5799.
- de Jong, H. (2002), Modeling and simulation of genetic regulatory systems : A literature review, *Journal of Computational Biology*, 9, 67–103.

- Dennunzio, A., Formenti, E., Manzoni, L., Mauri, G., Porreca, A. E. (2015), Computational complexity of finite asynchronous cellular automata, *Theoretical Computer Science*, .
- Dennunzio, A., Formenti, E., Manzoni, L., Porreca, A. E. (2019), Complexity of the dynamics of reaction systems, *Information and Computation*, 267, 96–109.
- Esparza, J., Heljanko, K. (2008), *Unfoldings : A Partial-Order Approach to Model Checking (Monographs in Theoretical Computer Science. An EATCS Series)*, 1 edition, Springer Publishing Company, Incorporated, Berlin, Heidelberg.
- Fages, F., Soliman, S. (2008), Abstract interpretation and types for systems biology, *Theoretical Computer Science*, 403(1), 52 – 70.
- Fatès, N. A. (2018), Asynchronous cellular automata, in R. Meyers, (ed.), *Encyclopedia of Complexity and Systems Science*, Springer, p. 21.
- Feret, J., Danos, V., Krivine, J., Harmer, R., Fontana, W. (2009), Internal coarse-graining of molecular systems, *Proceedings of the National Acedemy of Sciences of the United States of America*, 106(16).
- Formenti, E., Manzoni, L., Porreca, A. E. (2014), Fixed points and attractors of reaction systems, in *Language, Life, Limits*, Springer International Publishing, pp. 194–203.
- Glass, L., Kauffman, S. (1973), Logical analysis of continuous, non-linear biochemical control networks, *Journal of Theoretical Biology*, 39(1), 103–129.
- Haar, S., Kern, C., Schwoon, S. (2013), Computing the reveals relation in occurrence nets, *Theoretical Computer Science*, 493, 66–79.
- Hamez, A., Thierry-Mieg, Y., Kordon, F. (2009), Building efficient model checkers using hierarchical set decision diagrams and automatic saturation, *Fundam. Inf.*, 94(3-4), 413–437.
- Janicki, R., Kleijn, J., Koutny, M., Mikulski, Ł. (2015), Step traces, *Acta Informatica*, 53(1), 35–65.
- Kauffman, S. A. (1969), Metabolic stability and epigenesis in randomly connected nets, *Journal of Theoretical Biology*, 22, 437–467.
- Klarner, H., Bockmayr, A., Siebert, H. (2015), Computing maximal and minimal trap spaces of boolean networks, *Natural Computing*, 14(4), 535–544.
- Kolčák, J., Šafránek, D., Haar, S., Paulevé, L. (2018), Parameter Space Abstraction and Unfolding Semantics of Discrete Regulatory Networks, *Theoretical Computer Science*, 765, 120 – 144.
- Le Novère, N. (2015), Quantitative and logic modelling of molecular and gene networks., *Nature reviews. Genetics*, 16, 146–158.
- Mai, Z., Liu, H. (2009), Boolean network-based analysis of the apoptosis network : Irreversible apoptosis and stable surviving, *Journal of Theoretical Biology*, 259(4), 760 – 769.

- Mandon, H., Su, C., Pang, J., Paul, S., Haar, S., Paulevé, L. (2019), Algorithms for the Sequential Reprogramming of Boolean Networks, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(5), 1610–1619.
- Manzoni, L. (2012), Asynchronous cellular automata and dynamical properties, *Natural Computing*, 11(2), 269–276.
- Martínez-Sosa, P., Mendoza, L. (2013), The regulatory network that controls the differentiation of t lymphocytes, *Biosystems*, 113(2), 96 – 103.
- Melliti, T., Regnault, D., Richard, A., Sené, S. (2016), Asynchronous simulation of boolean networks by monotone boolean networks, in *Lecture Notes in Computer Science*, Springer International Publishing, pp. 182–191.
- Mizera, A., Pang, J., Qu, H., Yuan, Q. (2017), A new decomposition method for attractor detection in large synchronous boolean networks, in *Dependable Software Engineering. Theories, Tools, and Applications*, Springer International Publishing, pp. 232–249.
- Mizera, A., Pang, J., Qu, H., Yuan, Q. (2019), Taming asynchrony for attractor detection in large boolean networks, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(1), 31–42.
- Murata, T. (1989), Petri nets : properties, analysis and applications, *Proc. of the IEEE*, 77(4), 541–580.
- Nakamura, K. (1981), Synchronous to asynchronous transformation of polyautomata, *Journal of Computer and System Sciences*, 23(1), 22–37.
- Naldi, A. (2018), BioLQM : A java toolkit for the manipulation and conversion of logical qualitative models of biological networks, *Frontiers in Physiology*, 9.
- Naldi, A., Thieffry, D., Chaouiya, C. (2007), Decision diagrams for the representation and analysis of logical models of genetic networks, in *Proceedings of the 2007 international conference on Computational methods in systems biology, CMSB'07*, Springer-Verlag, Berlin, Heidelberg, pp. 233–247.
- Noual, M., Sené, S. (2017), Synchronism versus asynchronism in monotonic boolean automata networks, *Natural Computing*, 17(2), 393–402.
- Papadimitriou, C. H. (1995), *Computational Complexity*, Addison-Wesley.
- Paulevé, L. (2018), Reduction of Qualitative Models of Biological Networks for Transient Dynamics Analysis, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(4), 1167–1179.
- Paulevé, L., Kolčák, J., Chatain, T., Haar, S. (2020), Reconciling qualitative, abstract, and scalable modeling of biological networks, *Nature Communications*, 11(1).
- Petri, C. A. (1962), *Kommunikation mit Automaten*, PhD thesis, University of Bonn.
- Razaq, M., Paulevé, L., Siegel, A., Saez-Rodriguez, J., Bourdon, J., Guziolowski, C. (2018), Computational discovery of dynamic cell line specific boolean networks from multiplex time-course data, *PLOS Computational Biology*, 14(10), 1–23.

- Robert, F. (1995), *Les systemes dynamiques discrets*, vol. 19 of *Mathématiques et Applications*, Springer Science & Business Media.
- Rodríguez, C., Schwoon, S. (2013), Cufn : A tool for unfolding and verifying petri nets with read arcs, *in Automated Technology for Verification and Analysis*, Springer, pp. 492–495.
- Saez-Rodriguez, J., Alexopoulos, L. G., Epperlein, J., Samaga, R., Lauffenburger, D. A., Klamt, S., Sorger, P. K. (2009), Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction, *Molecular Systems Biology*, 5(1), 331.
- Snoussi, E. (1989), Qualitative dynamics of a piecewise-linear differential equations : a discrete mapping approach, *Dynamics and stability of Systems*, 4, 189–207.
- Su, C., Paul, S., Pang, J. (2019), Controlling large boolean networks with temporary and permanent perturbations, *in Lecture Notes in Computer Science*, Springer International Publishing, pp. 707–724.
- Thomas, R. (1973), Boolean formalization of genetic control circuits, *Journal of Theoretical Biology*, 42(3), 563 – 585.
- Thomas, R., d’Ari, R. (1990), *Biological Feedback*, CRC Press, Boca Raton , Florida, USA.
- Traynard, P., Fauré, A., Fages, F., Thieffry, D. (2016), Logical model specification aided by model-checking techniques : application to the mammalian cell cycle regulation, *Bioinformatics*, 32(17), i772–i780.
- Vogler, W., Semenov, A., Yakovlev, A. (1998), Unfolding and finite prefix for nets with read arcs, *in CONCUR’98 Concurrency Theory*, Springer Berlin Heidelberg, pp. 501–516.
- Wolfram, S. (1983), Statistical mechanics of cellular automata, *Rev. Mod. Phys.*, 55, 601–644.
- Zañudo, J. G., Steinway, S. N., Albert, R. (2018), Discrete dynamic network modeling of oncogenic signaling : Mechanistic insights for personalized treatment of cancer, *Current Opinion in Systems Biology*, 9, 1–10.
- Zañudo, J. G. T., Albert, R. (2015), Cell fate reprogramming by control of intracellular network dynamics, *PLOS Computational Biology*, 11(4), 1–24.