

# Analyses statiques de la dynamique des réseaux d'automates indéterministes

Loïc Paulevé, Maxime Folschette, Morgan Magnin, Olivier Roux

► **To cite this version:**

Loïc Paulevé, Maxime Folschette, Morgan Magnin, Olivier Roux. Analyses statiques de la dynamique des réseaux d'automates indéterministes. *Revue des Sciences et Technologies de l'Information - Série TSI: Technique et Science Informatiques*, Lavoisier, 2015, 34 (4), pp.22. <10.3166/TSI.34.463-484>. <hal-01070295v2>

**HAL Id: hal-01070295**

**<https://hal.archives-ouvertes.fr/hal-01070295v2>**

Submitted on 9 Sep 2015

**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.

---

# Analyses statiques de la dynamique des réseaux d'automates indéterministes

Loïc Paulevé<sup>1</sup>, Maxime Folschette<sup>2</sup>, Morgan Magnin<sup>2</sup>,  
Olivier Roux<sup>2</sup>

1. CNRS, Laboratoire de Recherche en Informatique (LRI)  
Université Paris-Sud - UMR CNRS 8623, France  
loic.pauleve@lri.fr

2. LUNAM Université, École Centrale de Nantes, IRCCyN UMR CNRS 6597  
(Institut de Recherche en Communications et Cybernétique de Nantes)  
1 rue de la Noë - B.P. 92101 - 44321 Nantes Cedex 3, France.  
{maxime.folschette,morgan.magnin,olivier.roux}@irccyn.ec-nantes.fr

---

*RÉSUMÉ.* Cet article présente une synthèse de résultats récents permettant l'analyse à grande échelle de la dynamique des réseaux d'automates. Nous nous focalisons sur deux types de propriétés: l'accessibilité – étant donné un état global du réseau, existe-t-il une suite de transitions permettant d'atteindre un état local donné pour un automate donné ; et la caractérisation des points fixes à partir de la topologie du réseau d'automates et de ses transitions locales. Pour l'accessibilité, nous donnons des conditions nécessaires applicables à tout réseaux d'automates et des conditions suffisantes pour les réseaux d'automates asynchrones. Nous abordons également le calcul d'ensembles de coupes pour l'accessibilité, qui sont des ensembles d'états locaux d'automates qui, si ils sont désactivés, rendent impossible l'accessibilité étudiée. Ces analyses de la dynamique exploitent la concurrence au sein des réseaux d'automates indéterministes et reposent sur des abstractions donnant une représentation compacte, non redondante, mais approchée, de l'ensemble des comportements possibles. Ces méthodes ont rendu possible l'analyse de la dynamique de réseaux biologiques avec des centaines et des milliers de composants.

*ABSTRACT.* This paper gives an overview of recent results on scalable analysis of automata networks dynamics. We focus on two properties: reachability – given a global state of the network, does it exist a sequence of transition reaching a given local state of a given automaton; and fixed point characterization. Concerning reachability, we give necessary conditions in the general scope of automata networks; and sufficient conditions in the scope of asynchronous automata networks. We address the computation of cut sets for reachability, that are sets of local states, that if disabled make impossible the concerned reachability. Those analyses exploit the concurrency in indeterministic automata networks and rely on abstractions giving compact, non-redundant, but approximate representations of the network dynamics. Such methods have made tractable the analysis of dynamics of biological networks gathering hundreds and thousands of components.

## 1. Introduction

Un système concurrent complexe est caractérisé par le fait que sa dynamique résulte des interactions entre les composants élémentaires qui constituent ce système. Ces interactions entre composants sont souvent de nature très simple au niveau élémentaire, et la complexité provient essentiellement de la combinaison des influences positive ou négative que ces éléments ont les uns sur les autres. En outre, chacune de ces influences peut d'ailleurs évoluer dans le temps en type et en importance, en fonction d'autres influences qu'elles subissent également. Or, la complexité induite par ces combinaisons et ces évolutions peut être très grande. Les modélisations des systèmes ayant un grand nombre de composants aboutissent alors à des objets qui, pour rendre compte assez précisément du fonctionnement des systèmes qu'ils représentent, peuvent atteindre des tailles gigantesques. Les algorithmes et programmes d'analyse des comportements dans ces modèles sont donc souvent inapplicables car les questions auxquels ils devraient apporter des réponses ne sont pas traitables en temps ou en espace dues à leur complexité (généralement PSPACE-dur).

Dans le contexte de la biologie systémique, René Thomas a étudié les relations entre les structures logiques des systèmes et leur capacité à générer des états stationnaires multiples ou des oscillations entretenues (Thomas, 1973), et il a formalisé les fonctionnements des systèmes par réseaux booléens asynchrones (Thomas, 1981). Suivant la même approche, l'influence des rétroactions positives ou négatives sur la stabilité d'un système est plus généralement discutée dans (Cinquin, Demongeot, 2002) puis récapitulée de façon très complète dans (Comet *et al.*, 2013). Ainsi, les influences réciproques entre les composants de tels systèmes dynamiques étant souvent représentées par un graphe d'interactions, des caractérisations de réseaux booléens asynchrones ont pu être identifiées pour distinguer différentes classes de fonctionnalités des circuits de rétroaction à partir de ces graphes. Ces fonctionnalités induisent différents types de comportement : couramment, l'existence d'attracteurs cycliques ou la présence de points fixes multiples, ce qui revient à des phénomènes d'oscillations entretenues ou de multistationnarité.

Bien que certains résultats utiles sur des caractéristiques du fonctionnement des systèmes complexes aient ainsi été obtenus par des études structurelles ou topologiques des réseaux booléens, de telles représentations ne fournissent pas toujours assez de détails pour étudier plus à fond des propriétés dynamiques fines, et notamment pour conduire une analyse plus complète dans les cas de perturbations et d'irrégularités. On souhaiterait même aller plus loin en découvrant comment modifier l'émergence de comportements caractéristiques, par exemple pour éviter d'atteindre des états non souhaitables

Ainsi, par exemple, les travaux visant à rechercher les modifications structurelles qui permettraient d'éviter certaines évolutions néfastes imposent de compléter les modélisations. C'est dans cette optique que les graphes de régulation multivalués ont été étudiés en se basant sur des représentations permettant de surcroît de mieux refléter d'autres aspects telles que les combinaisons logiques des interactions ou les niveaux à partir desquels les interactions deviennent effectives. Ces outils formels additionnels sont, entre autres, les réseaux de Petri ou encore les réseaux d'automates finis communicants.

Plus particulièrement, l'analyse des propriétés dynamiques pour des systèmes de régulation biologique est étudiée dans (Naldi *et al.*, 2007) grâce à l'introduction de *diagrammes de décision multivalués*. Ceux-ci permettent d'exprimer les fonctions logiques qui accompagnent le graphe de régulation en régissant les interactions élémentaires et précisant ainsi les comportements effectifs. Ce type de représentation permet de développer des algorithmes performants qui conduisent à des résultats sur l'identification d'états stables ainsi que ce que les auteurs appellent des contextes fonctionnels de circuits de régulation, induisant les propriétés dynamiques qui sont recherchées. La même équipe complète cette approche dans (Chaouiya *et al.*, 2011) en s'appuyant toujours sur ces diagrammes de décision multivalués pour représenter les paramètres logiques et afin de traduire le graphe de régulation en un réseau de Petri équivalent sur lequel ils sont capables d'étudier encore plus efficacement les comportements. Pour compacter la représentation de façon à n'avoir qu'une seule place par composant biologique, ils appliquent en outre une approche portant sur une représentation par réseaux de Petri colorés.

Également basées sur différentes formalisations logiques qui sont fréquemment des réseaux de Petri, des analyses d'un type un peu analogue sont effectuées pour des problèmes classiques de diagnostic de systèmes matériels. Il s'agit alors de retrouver quel a pu être le comportement effectif d'un système ayant abouti à l'occurrence d'un événement observable de type alarme. Afin d'assurer un fonctionnement amélioré, il faut ensuite être capable d'éviter que ce comportement puisse se produire. Ce type de problème consiste donc également à repérer, par les analyses qui conviennent, certains fonctionnements pernicious — sous forme de traces d'exécution — et plus ou moins invisibles dans le modèle représentant toutes les exécutions possibles du système. Comme le nombre d'exécutions peut être immense, spécialement dans des systèmes hautement parallélisés, il est indispensable d'appliquer des démarches bien adaptées à ces systèmes dynamiques complexes. Plusieurs travaux s'y emploient, notamment en considérant le *dépliage de Réseaux de Petri*. C'est ce qui est fait par exemple dans (Baldan *et al.*, 2010). Le principe de leur approche est basé sur la réécriture de graphes et son efficacité tient à la détermination d'exécutions partiellement ordonnées, plutôt que des entrelacements explosifs, pour enregistrer et traiter les traces extraites de leurs modèles.

### *Analyses statiques de la dynamique*

Dans cet article, nous présentons une synthèse de résultats pour l'analyse de la dynamique de réseaux d'automates applicables à très grande échelle, c.-à-d. composés de centaines voire de milliers de composants. En particulier, nous nous concentrons ici sur les propriétés d'accessibilité et de points fixes. Comme nous le soulignerons en section 6, ces analyses ont permis d'appréhender l'étude de la dynamique de réseaux biologiques d'une taille jusqu'alors hors d'atteinte par les méthodes classiques de vérification formelle.

De manière analogue aux dépliages de réseaux de Petri, nous exploitons une description de la dynamique en termes de transitions définies localement entre les états des composants en interaction. Cette description locale des transitions met en exergue la concurrence entre le changement des états des composants, et, comme exploité par les dépliages, permet d'ignorer un grand nombre d'exécutions non informatives sur la dynamique du réseau.

La spécificité de notre approche repose sur une interprétation abstraite des comportements concurrents des réseaux d'automates. À partir de la spécification du réseau d'automates, nous calculons des représentations abstraites de l'ensemble des comportements concernés par la propriété d'accessibilité recherchée. Ces représentations prennent la forme de graphes, que nous appelons Graphes de Causalité Locale (GCL). Nos abstractions oublient délibérément une partie de l'information sur l'ordre ou l'arité des transitions locales, résultant ainsi en des approximations supérieures et inférieures des comportements du modèle concret. Les résultats présentés dans les sections 3 et 4 identifient alors des propriétés au sein de ces GCL qui sont soit nécessaires, soit suffisantes, à l'accessibilité étudiée.

Cette méthode a comme avantage majeur une complexité très réduite comparée à une vérification formelle exacte : nos approches sont généralement exponentielles selon le nombre d'états au sein d'un seul automate, mais polynomiales selon le nombre d'automates. Ainsi, nous garantissons leur applicabilité pour l'étude de la dynamique de très grands réseaux où chaque automate n'a que peu d'états locaux – ce qui est typiquement le cas des modèles qualitatifs des réseaux biologiques. L'inconvénient de l'approche par approximation est bien sûr la possibilité d'obtenir une réponse non concluante pour le modèle concret, appelant alors à raffiner l'analyse de la dynamique.

Les résultats que nous présentons ont tous été obtenus dans le cadre des Frappes de Processus (Paulevé, 2011), une classe particulière des réseaux d'automates asynchrones où toutes les interactions sont décomposées en interactions binaires (entre deux automates). Pour cette synthèse, nous généralisons ces résultats à différentes classes de réseaux d'automates : à tout réseau d'automates pour les conditions nécessaires pour l'accessibilité ; aux réseaux d'automates asynchrones pour les conditions suffisantes ; aux Frappes de Processus pour la caractérisation des points fixes.

## Plan

La définition générale des réseaux d'automates indéterministes, centrée sur les transitions locales, est présentée en section 2 avec les différentes classes (asynchrones, frappes de processus) considérées dans cet article. La section 3 introduit la notion de Graphe de Causalité Locale, et établit des conditions nécessaires à l'accessibilité dans la classe générale des réseaux d'automates. Nous abordons également dans cette section le calcul de points de passage obligés pour atteindre un état donné, et permettant ainsi de contrôler cette accessibilité. La section 4 établit des conditions suffisantes à l'accessibilité pour les réseaux d'automates asynchrones. La section 5 donne une caractérisation des points fixes dans la classe des frappes de processus, réduisant la recherche de points fixes à la recherche de cliques dans une représentation graphique du réseau d'automates. Enfin, nous discutons en section 6 des applications de ces résultats pour la modélisation et l'analyse des grands réseaux biologiques.

## Notations

Étant donné un ensemble fini  $A$ , son cardinal est noté  $|A|$ . L'ensemble d'entiers  $\{a, a+1, \dots, a+n\}$  est dénoté par  $[a; a+n]$ . Étant donné des ensembles d'ensembles  $B^1, \dots, B^n \in \wp(\wp(A))$ ,  $\prod_{i \in [1; n]} B^i \triangleq B^1 \tilde{\times} \dots \tilde{\times} B^n \in \wp(\wp(A))$  est le produit d'ensembles d'ensembles où  $\{e_1, \dots, e_n\} \tilde{\times} \{e'_1, \dots, e'_m\} \triangleq \{e_i \cup e'_j \mid i \in [1; n] \wedge j \in [1; m]\}$ . En particulier,  $\forall (i, j) \in [1; n] \times [1; m]$ ,  $B^i \tilde{\times} B^j = B^j \tilde{\times} B^i$  et  $\emptyset \tilde{\times} B^i = \emptyset$ . Le produit vide est défini par  $\prod_{\emptyset} \triangleq \{\emptyset\}$ . Les graphes sont définis par un couple  $(V, E)$  où  $V$  sont les nœuds et  $E$  les arêtes. Pour les graphes dirigés (resp. non dirigés),  $E$  est un ensemble de couples (resp. d'ensembles de cardinal deux) de nœuds. Dans le cas d'un graphe dirigé  $(V, E)$ , les enfants d'un nœud  $n$  sont donnés par la fonction  $\text{children} : V \rightarrow \wp(V)$ , où  $\text{children}(n) \triangleq \{m \in V \mid (n, m) \in E\}$ ; ses parents par la fonction  $\text{parents} : V \rightarrow \wp(V)$  où  $\text{parents}(n) \triangleq \{m \in N \mid (m, n) \in E\}$ ; et ses descendants par  $\text{conn}_{(V, E)}(n)$  qui est le plus petit point fixe contenant  $n$  de la fonction  $\text{fconn} : \wp(V) \rightarrow \wp(V)$  où  $\text{fconn}(W) \triangleq \bigcup_{m \in W} \text{children}(m)$ .

## 2. Réseaux d'automates finis indéterministes

Un réseau d'automates (définition 1) est formé par un ensemble d'automates finis où chaque transition au sein d'un automate est associée à un libellé. Un libellé peut être partagé entre plusieurs transitions d'automates différents. Ce libellé ( $\ell$ ) définit alors une pré-condition ( $\bullet\ell$ ) sur l'état global du réseau : tout automate partageant le libellé doit être dans l'état permettant la transition associée. Lorsqu'une telle condition est satisfaite, chaque automate change d'état en suivant les transitions associées; l'état global du réseau contient alors tous les états locaux des automates résultant des transitions associées à  $\ell$ , référencés dans la post-condition  $\ell\bullet$ . Lorsque plusieurs pré-conditions sont satisfaites par un même état global, seul un libellé est choisi, de manière non déterministe. Cette définition est similaire aux réseaux de Petri *saufs* (au

plus un jeton par place) (Bernardinello, De Cindio, 1992) avec des groupes de places mutuellement exclusives (délimitant les automates).

**DÉFINITION 1** (Réseau d'automates  $(\Sigma, S, \mathcal{L}, T)$ ). — *Un réseau d'automates est défini par un quadruplet  $(\Sigma, S, \mathcal{L}, T)$  où*

- $\Sigma = \{a, b, \dots, z\}$  est un ensemble fini de noms d'automates ;
- Pour tout  $a \in \Sigma$ ,  $S(a) = [1; k_a]$  est l'ensemble fini des états locaux de l'automate  $a$  ;  $S = \prod_{a \in \Sigma} [1; k_a]$  est l'ensemble fini des états globaux du réseau ;
- $\mathcal{L} = \{\ell_1, \dots, \ell_m\}$  est l'ensemble fini des libellés de transitions ;
- $T = \{a \mapsto T_a \mid a \in \Sigma\}$ , où  $\forall a \in \Sigma, T_a \subset [1; k_a] \times \mathcal{L} \times [1; k_a]$ , associe chaque automate à un ensemble fini de transitions entre états locaux.

*Notations.* Nous notons  $i \xrightarrow{\ell} j \in T(a) \triangleq (i, \ell, j) \in T_a$  et  $a_i \xrightarrow{\ell} a_j \in T \triangleq i \xrightarrow{\ell} j \in T(a)$ . Pour tout libellé  $\ell \in \mathcal{L}$ , nous notons  $\bullet\ell \triangleq \{a_i \mid i \xrightarrow{\ell} j \in T(a)\}$  et  $\ell\bullet \triangleq \{a_j \mid i \xrightarrow{\ell} j \in T(a)\}$ . La réunion des états locaux des automates est dénotée par  $\mathbf{LS} \triangleq \{a_i \mid a \in \Sigma \wedge i \in S(a)\}$ . Enfin, étant donné un état global  $s \in S$ ,  $s(a) \in S(a)$  fait référence à l'état local de l'automate  $a \in \Sigma$ .

La définition 2 formalise la relation binaire de transition entre deux états globaux d'un réseau d'automates.

**DÉFINITION 2** (Sémantique des réseaux d'automates). — *Étant donné un réseau d'automates  $(\Sigma, S, \mathcal{L}, T)$ , la relation de transition globale entre deux états du réseau  $\rightarrow \subset S \times S$  est définie par :*

$$s \rightarrow s' \triangleq \exists \ell \in \mathcal{L} : \forall a_i \in \bullet\ell, s(a) = i \wedge \forall a_j \in \ell\bullet, s'(a) = j \\ \wedge \forall b \in \Sigma, S(b) \cap \bullet\ell = \emptyset \Rightarrow s(b) = s'(b).$$

**REMARQUE.** — Cette définition des réseaux d'automates diffère de la définition généralement utilisée dans les travaux sur les réseaux booléens ou multi-valués (p. ex. Remy *et al.*, 2008 ; Aracena *et al.*, 2009 ; Richard, 2010 ; Noual *et al.*, 2012) où chaque automate  $a$  d'un réseau se définit par une fonction *déterministe*  $f^a$  qui associe à l'état global du réseau l'état suivant de l'automate  $a$ . Dans cet article, notre modélisation de la dynamique permet d'encoder un non-déterminisme sur l'évolution d'un composant dans un contexte donné. Par exemple, pour un état global du réseau, il est possible de définir que le composant  $a$  peut, de manière non déterministe, soit augmenter, soit diminuer. En particulier, il est autorisé d'avoir deux transitions  $a_i \xrightarrow{\ell_1} a_j$  et  $a_i \xrightarrow{\ell_2} a_k$  avec la même pré-condition  $\bullet\ell_1 = \bullet\ell_2$  mais des post-conditions différentes  $\ell_1\bullet \neq \ell_2\bullet$ .  $\square$

La définition des réseaux d'automates présentée ici permet d'encoder la dynamique des réseaux booléens et multi-valués, où la mise à jour des automates peut être asynchrone ou synchrone.

Certains résultats présentés dans cet articles reposent sur des restrictions de cette définition générale : les réseaux d'automates asynchrones (définition 3) où seul un automate change d'état lors d'une transition globale ; et les frappes de processus (définition 4) où un seul automate change d'état lors d'une transition globale, et où un libellé de transition locale est partagé par au plus deux automates.

**DÉFINITION 3 (Réseau d'automates asynchrone).** — *Un réseau d'automates  $(\Sigma, S, \mathcal{L}, T)$  est un réseau d'automates asynchrone si et seulement si, pour tout  $\ell \in \mathcal{L}$ ,  $|\{a_j \mid a_i \xrightarrow{\ell} a_j, a_i \neq a_j\}| = 1$ .*

**DÉFINITION 4 (Frappes de processus).** — *Un réseau d'automates  $(\Sigma, S, \mathcal{L}, T)$  est une frappe de processus si et seulement si  $(\Sigma, S, \mathcal{L}, T)$  est un réseau d'automates asynchrone et, pour tout  $\ell \in \mathcal{L}$ ,  $|\bullet\ell| \leq 2$ .*

Étant donné un réseau d'automates  $\mathcal{N} = (\Sigma, S, \mathcal{L}, T)$  et un sous-ensemble de ses états locaux  $ls \subseteq \mathbf{LS}$ ,  $\mathcal{N} \ominus ls$  fait référence au réseau d'automates où tous les états locaux présents dans  $ls$  ont été désactivés, c.-à-d. où toutes les transitions les impliquant ont été supprimées (définition 5).

**DÉFINITION 5 (Désactivation d'états locaux).** — *Étant donné  $\mathcal{N} = (\Sigma, S, \mathcal{L}, T)$  et  $ls \in \wp(\mathbf{LS})$ ,  $\mathcal{N} \ominus ls \triangleq (\Sigma, S, \mathcal{L}', T')$  où  $\mathcal{L}' = \{\ell \in \mathcal{L} \mid ls \cap \bullet\ell = \emptyset\}$  et  $T' = \{a_j \xrightarrow{\ell} a_j \in T \mid \ell \in \mathcal{L}'\}$ .*

**EXEMPLE 6.** — Soit le réseau d'automates  $(\Sigma, S, \mathcal{L}, T)$  défini par

$$\begin{array}{ll} \Sigma \triangleq \{a, b, c, d\} & \mathcal{L} \triangleq \{\ell_1, \ell_2, \ell_3, \ell_4, \ell_5, \ell_6\} \\ S(a) \triangleq [1; 3] & T(a) \triangleq \{1 \xrightarrow{\ell_2} 2; 2 \xrightarrow{\ell_3} 3; 1 \xrightarrow{\ell_1} 3; 3 \xrightarrow{\ell_4} 2\} \\ S(b) \triangleq [1; 3] & T(b) \triangleq \{1 \xrightarrow{\ell_2} 2; 1 \xrightarrow{\ell_5} 3; 1 \xrightarrow{\ell_6} 1; 3 \xrightarrow{\ell_1} 2\} \\ S(c) \triangleq [1; 2] & T(c) \triangleq \{1 \xrightarrow{\ell_4} 2; 2 \xrightarrow{\ell_3} 1\} \\ S(d) \triangleq [1; 2] & T(d) \triangleq \{1 \xrightarrow{\ell_6} 2; 2 \xrightarrow{\ell_5} 1\} . \end{array}$$

Ce réseau n'est pas un réseau d'automates asynchrone : seule la transition  $\ell_6$  est asynchrone, les autres transitions  $(\ell_1, \dots, \ell_5)$  induisent le changement simultané de plusieurs automates.

Partant de l'état  $\langle 1, 1, 1, 1 \rangle$  – les automates sont dans l'ordre  $a, b, c, d$  –, la relation de transition globale donne l'ensemble des comportements suivants :

$$\begin{array}{ccccccc} \langle 1, 1, 1, 1 \rangle & \xrightarrow{\ell_6} & \langle 1, 1, 1, 2 \rangle & \xrightarrow{\ell_5} & \langle 1, 3, 1, 1 \rangle & \xrightarrow{\ell_1} & \langle 3, 2, 1, 1 \rangle \\ & & \downarrow \ell_2 & & \downarrow \ell_2 & & \downarrow \ell_3 \left( \begin{array}{c} \uparrow \\ \downarrow \end{array} \right) \ell_4 \\ \langle 2, 2, 1, 1 \rangle & & \langle 2, 2, 1, 2 \rangle & & & & \langle 2, 2, 2, 1 \rangle \end{array}$$

□



### 3. Conditions nécessaires pour l'accessibilité dans les réseaux d'automates

Dans cette section, nous nous intéressons aux conditions nécessaires à l'existence d'une suite de transitions au sein d'un réseau  $(\Sigma, S, \mathcal{L}, T)$  partant d'un état global initial  $s \in S$  et atteignant un état  $s' \in S$  où  $s(a) = j$  pour un automate  $a \in \Sigma$  et un état local  $j \in S(a)$  donnés. Dans la suite, nous généralisons la notion d'état global par la notion de contexte (définition 7) : un contexte délimite pour chaque automate un ensemble d'états locaux initiaux possibles. Étant donné un contexte  $\varsigma$  et un état local  $a_j$  à atteindre, nous cherchons donc des conditions nécessaires pour qu'il existe un état  $s$  compatible avec  $\varsigma$  tel que  $a_j$  est accessible depuis  $s$ .

**DÉFINITION 7 (Contexte  $\varsigma$ ).** — *Étant donné un réseau d'automates  $(\Sigma, S, \mathcal{L}, T)$ , un contexte  $\varsigma$  est une association entre chaque automate  $a \in \Sigma$  et un sous-ensemble non vide de ses états locaux :  $\forall a \in \Sigma, \varsigma(a) \in \wp(S(a)) \wedge \varsigma(a) \neq \emptyset$ .*

Les résultats présentés dans cette section reposent sur une interprétation abstraite de la dynamique des réseaux d'automates introduite par Paulevé *et al.* (2012). Le principe est de calculer, à partir du réseau, un graphe qui va contenir des informations sur les étapes intermédiaires nécessaires à l'accessibilité d'un état local donné. Le graphe obtenu est appelé graphe de causalité locale. Nous donnons ici des conditions nécessaires sur la topologie de ce graphe pour que l'accessibilité soit satisfaite.

#### 3.1. Graphe de causalité locale

Nous appelons *objectif* (définition 8) l'accessibilité de  $a_j$  depuis  $a_i$ , noté  $a_i \rightarrow^* a_j$ .

**DÉFINITION 8 (Objectif).** — *Étant donné un réseau d'automates  $(\Sigma, S, \mathcal{L}, T)$ , l'accessibilité de l'état local  $a_j$  depuis  $a_i$  est appelé un objectif et est noté  $a_i \rightarrow^* a_j$ . L'ensemble de tous les objectifs d'un réseau d'automates est dénoté par  $\mathbf{Obj} \triangleq \{a_i \rightarrow^* a_j \mid (a_i, a_j) \in \mathbf{LS} \times \mathbf{LS}\}$ .*

Pour un objectif  $P = a_i \rightarrow^* a_j \in \mathbf{Obj}$  donné, nous définissons  $\text{sol}(P) \subset \wp(\mathbf{LS})$  la *causalité locale* de  $P$  (définition 9) : chaque  $ls \in \text{sol}(P)$  est un ensemble d'états locaux qui peuvent être impliqués dans les transitions permettant l'accessibilité de  $a_j$  depuis  $a_i$ . Nous appelons  $ls$  une *solution* pour l'objectif  $P$ .

**DÉFINITION 9 (Causalité locale).** —  *$\text{sol} : \mathbf{Obj} \rightarrow \wp(\wp(\mathbf{LS}))$  associe à un objectif un ensemble d'ensembles d'états locaux (solutions) tel que  $\forall P \in \mathbf{Obj}, \forall ls \in \text{sol}(P), \nexists ls' \in \text{sol}(P), ls \neq ls' \text{ avec } ls' \subset ls$ . L'ensemble de ces associations est noté  $\mathbf{Sol} \triangleq \{P, ls \mid ls \in \text{sol}(P)\}$ .*

Les ensembles de solutions  $\text{sol}(P)$  sont dits nécessaires si la désactivation d'au moins un état local de *chaque* solution rend l'accessibilité de  $a_j$  impossible depuis n'importe quel état contenant  $a_i$  (définition 10). Ainsi, si  $\text{sol}(P) = \{\{b_k\} \cup ls^1, \dots, ls^m\}$  est nécessaire alors  $\text{sol}'(P) = \{ls^1, \dots, ls^m\}$  est également nécessaire ;  $\text{sol}(a_i \rightarrow^* a_j) = \emptyset$  implique que  $a_j$  n'est jamais accessible depuis  $a_i$ , et  $\forall a_i \in \mathbf{LS}, \text{sol}(a_i \rightarrow^* a_i) \triangleq \{\emptyset\}$ .

**DÉFINITION 10** (sol nécessaire). —  $\text{sol}(a_i \rightarrow^* a_j) = \{ls^1, \dots, ls^n\}$  est un ensemble nécessaire de solutions pour le réseau d'automates  $\mathcal{N} = (\Sigma, S, \mathcal{L}, T)$  si et seulement si  $\forall kls \in \prod_{i \in [1;n]} \{ls^i\}$ ,  $a_j$  n'est pas accessible dans le réseau  $\mathcal{N} \ominus kls$  depuis tout état  $s \in S$  où  $s(a) = i$ .

Étant donné un réseau d'automates  $(\Sigma, S, \mathcal{L}, T)$  et un objectif  $a_i \rightarrow^* a_j \in \mathbf{Obj}$ , nous pouvons calculer un ensemble de solutions nécessaires  $\text{sol}(a_i \rightarrow^* a_j)$  de la façon suivante : pour chaque séquence acyclique  $i \xrightarrow{\ell_1} \dots \xrightarrow{\ell_m} j$  de transitions locales dans  $T(a)$ , en définissant  $\text{ext}_a(\ell) \triangleq \{b_j \in \mathbf{LS} \mid b_j \xrightarrow{\ell} b_k \in T, b \neq a\}$ , nous posons  $ls \in \prod_{\ell \in \{\ell_1, \dots, \ell_m \mid \text{ext}_a(\ell) \neq \emptyset\}} \{\text{ext}_a(\ell)\} \Rightarrow ls \in \text{sol}(a_i \rightarrow^* a_j)$ , à la suppression des sur-ensembles près. La condition de la définition 10 se vérifie aisément avec cette construction : chaque  $kls \in \prod_{i \in [1;n]} \{ls^i\}$  intersecte tous les chemins locaux au sein de l'automate  $a$  entre les états  $i$  et  $j$  ; ainsi, désactiver un  $kls$  enlève toute possibilité d'atteindre  $a_j$  depuis  $a_i$ .

Par la définition 10, si  $a_j$  est accessible depuis un état  $s$  où  $s(a) = i$ , il existe  $ls \in \text{sol}(a_i \rightarrow^* a_j)$  tel que, pour tout  $b_k \in ls$ ,  $b_k$  est accessible depuis  $s$ . Le *Graphe (dirigé) de Causalité Locale* (GCL, définition 11) représente ce raisonnement récursif en liant chaque état local  $a_j$  vers les objectifs  $a_i \rightarrow^* a_j$ ,  $a_i \in \varsigma(a)$  ; chaque objectif  $P \in \mathbf{Obj}$  vers ses solutions  $\langle P, ls \rangle \in \mathbf{Sol}$  ; chaque solution  $\langle P, ls \rangle$  vers les états locaux qui la composent. Un GCL peut bien sûr contenir des cycles.

**DÉFINITION 11** (Graphe de causalité locale). — Étant donné un réseau d'automates  $(\Sigma, S, \mathcal{L}, T)$  avec un contexte  $\varsigma$  et un état local  $\omega \in \mathbf{LS}$ , le graphe de causalité locale (GCL)  $(V_\varsigma^\omega, E_\varsigma^\omega)$ , avec  $V_\varsigma^\omega \subseteq \mathbf{LS} \cup \mathbf{Obj} \cup \mathbf{Sol}$  et  $E_\varsigma^\omega \subseteq V_\varsigma^\omega \times V_\varsigma^\omega$ , est la plus petite structure satisfaisant :

1.  $\omega \in V_\varsigma^\omega$
2.  $a_i \in V_\varsigma^\omega \cap \mathbf{LS} \Leftrightarrow \{(a_i, a_j \rightarrow^* a_i) \mid a_j \in \varsigma\} \subseteq E_\varsigma^\omega$
3.  $P \in V_\varsigma^\omega \cap \mathbf{Obj} \Leftrightarrow \{(P, \langle P, ls \rangle) \mid \langle P, ls \rangle \in \mathbf{Sol}\} \subseteq E_\varsigma^\omega$
4.  $\langle P, ls \rangle \in V_\varsigma^\omega \cap \mathbf{Sol} \Leftrightarrow \{(\langle P, ls \rangle, a_i) \mid a_i \in ls\} \subseteq E_\varsigma^\omega$ .

La construction du GCL peut ainsi se faire de manière itérative, en commençant par l'état local  $\omega$  : lorsque qu'un état local  $a_i$  est ajouté, un objectif est ajouté à ses enfants pour chaque état local initial possible (spécifié dans  $\varsigma(a)$ ) ; lorsque un objectif  $P$  est ajouté, une solution est ajoutée à ses enfants pour chaque ensemble d'états locaux  $ls \in \text{sol}(P)$  ; chaque solution a pour enfants les états locaux qui la compose.

Dans le pire des cas,  $|V_\varsigma^\omega| = |\mathbf{LS}| + |\mathbf{Obj}| + |\mathbf{Sol}|$  et  $|E_\varsigma^\omega| = |\mathbf{Obj}| + |\mathbf{Sol}| + \sum_{\langle P, ls \rangle \in \mathbf{Sol}} |ls|$  où  $|\mathbf{LS}| = \sum_{a \in \Sigma} |S(a)|$  et  $|\mathbf{Obj}| = \sum_{a \in \Sigma} |S(a)|^2$ . Le nombre de solutions par objectif correspond au nombre de séquences acycliques de transitions locales au sein d'un automate qui sont donc bornées par le nombre d'états locaux de l'automate :  $|\mathbf{Sol}| \leq \sum_{a \in \Sigma} \binom{|T(a)|}{|S(a)|}$ . Ainsi, la complexité totale est exponentielle selon le nombre d'états locaux au sein d'un seul automate, et polynomiale selon le nombre d'automates et le nombre de transitions locales.

EXEMPLE 12. — La figure 1 montre le GCL pour le réseau d'automates de l'exemple 6 pour l'accessibilité de  $a_3$  depuis le contexte  $\varsigma = \{a \mapsto \{1\}; b \mapsto \{1\}; c \mapsto \{1, 2\}; d \mapsto \{1\}\}$ . Les états locaux sont représentés avec des carrés, et les éléments de  $\mathbf{Sol}$  par des cercles.

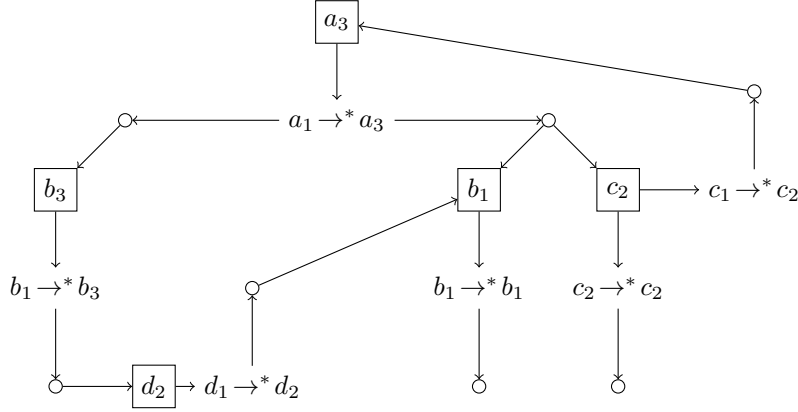


Figure 1. Graphe de Causalité Locale pour l'accessibilité de  $a_3$  dans le réseau d'automate de l'exemple 6

Pour l'automate  $a$ , il existe deux séquences acycliques pour aller de 1 vers 3 :  $1 \xrightarrow{\ell_2} 2 \xrightarrow{\ell_3} 3$  et  $1 \xrightarrow{\ell_1} 3$ . Ainsi, si  $a_3$  est atteint depuis  $a_1$ , alors nécessairement, une de ces deux séquences a été utilisée (mais pas forcément de manière consécutive). Pour chacune de ces transitions, le libellé est partagé avec exactement un état local d'un autre automate :  $b_1, c_2, b_3$  pour  $\ell_2, \ell_3, \ell_1$ , respectivement. Donc, si  $a_3$  est atteint depuis  $a_1$ , alors nécessairement, soit  $b_1$  et  $c_2$ , ou soit  $b_3$  ont été atteints avant. D'où  $\text{sol}(a_1 \rightarrow^* a_3) = \{\{b_1, c_2\}, \{b_3\}\}$ .  $\square$

### 3.2. Motif du GCL nécessaire pour l'accessibilité

Étant donné un réseau d'automates  $(\Sigma, S, \mathcal{L}, T)$ , un contexte  $\varsigma$  et un état local  $a_j \in \mathbf{LS}$ , nous pouvons déduire des définitions 10 et 11 que l'accessibilité de  $a_j$  depuis un état dans  $\varsigma$  implique l'existence d'un parcours du GCL partant du nœud  $a_j$  et tel que : si un nœud de type état local est traversé, alors au moins un de ses enfants (objectifs) est traversé ; si un nœud de type objectif est traversé, alors au moins un de ses enfants (solutions) est traversé ; si un nœud de type solution est traversé, alors tous ses enfants (états locaux) sont traversés ; le parcours est sans cycle. L'ensemble des solutions et états locaux à partir desquels un tel parcours existe dans le GCL  $(V_\varsigma^\omega, E_\varsigma^\omega)$ , où  $\omega = a_j$ , est donné par  $\text{validNodes}(V_\varsigma^\omega, E_\varsigma^\omega)$  (définition 13).

DÉFINITION 13. — Étant donné un GCL  $(V_\varsigma^\omega, E_\varsigma^\omega)$ ,  $\text{validNodes}(V_\varsigma^\omega, E_\varsigma^\omega) \subset V_\varsigma^\omega$  est le plus petit ensemble satisfaisant :

1.  $\{(P, \emptyset) \in V_\varsigma^\omega \cap \mathbf{Sol}\} \subset \text{validNodes}(V_\varsigma^\omega, E_\varsigma^\omega)$
2.  $\langle a_i \rightarrow^* a_j, ls \rangle \in \text{validNodes}(V_\varsigma^\omega, E_\varsigma^\omega) \implies a_j \in \text{validNodes}(V_\varsigma^\omega, E_\varsigma^\omega)$

3.  $\forall P \in V_\zeta^\omega \cap \mathbf{Obj}$  où  $\text{sol}(P) \neq \emptyset$ ,  $\forall ls \in \text{sol}(P)$ ,  $ls \subset \text{validNodes}(V_\zeta^\omega, E_\zeta^\omega) \implies \langle P, ls \rangle \in \text{validNodes}(V_\zeta^\omega, E_\zeta^\omega)$ .

Le théorème 14 établit une condition nécessaire pour l'accessibilité d'un état local à partir d'un contexte donné.

**THÉORÈME 14** (Condition nécessaire pour l'accessibilité). — Soit  $(\Sigma, S, \mathcal{L}, T)$  un réseau d'automates,  $\omega = a_j \in \mathbf{LS}$  un de ses états locaux, et  $\zeta$  un contexte pour le réseau.  $a_j$  est accessible depuis un état délimité par  $\zeta$  seulement si  $a_j \in \text{validNodes}(V_\zeta^\omega, E_\zeta^\omega)$ , où  $(V_\zeta^\omega, E_\zeta^\omega)$  est le Graphe de Causalité Locale, et  $\forall P \in V_\zeta^\omega \cap \mathbf{Obj}$ ,  $\text{sol}(P)$  est nécessaire.

**PREUVE.** — Pour toute séquence de transitions globales  $s^1 \rightarrow \dots \rightarrow s^n$  avec  $s^1(a) = i$ ,  $s^n(a) = j$  et  $\forall b \in \Sigma$ ,  $s^1(b) \in \zeta(b)$ , d'après la définition 10, il existe une solution  $ls \in \text{sol}(a_i \rightarrow^* a_j)$  telle que :

- soit  $ls = \emptyset$ , et par définition 13,  $\langle a_i \rightarrow^* a_j, \emptyset \rangle \in \text{validNodes}(V_\zeta^\omega, E_\zeta^\omega)$ , et  $a_j \in \text{validNodes}(V_\zeta^\omega, E_\zeta^\omega)$ ;
- soit  $\forall c_k \in ls$ , il existe  $m$ ,  $1 \leq m < n$ , où  $s^m(c) = k$ , et par induction, comme  $ls \subset \text{validNodes}(V_\zeta^\omega, E_\zeta^\omega)$ ,  $\langle a_i \rightarrow^* a_j, ls \rangle \in \text{validNodes}(V_\zeta^\omega, E_\zeta^\omega)$ , d'où  $a_j \in \text{validNodes}(V_\zeta^\omega, E_\zeta^\omega)$ .

■

Des conditions nécessaires supplémentaires pour une séquence d'accessibilités (atteindre  $a_j$  puis  $b_k$ , etc.) peuvent également être dérivées à partir du GCL et sont détaillées dans (Paulevé *et al.*, 2012).

**EXEMPLE 15.** — L'accessibilité de  $a_3$  dans le réseau d'automates de l'exemple 6 depuis le contexte  $\zeta = \{a \mapsto \{1\}; b \mapsto \{1\}; c \mapsto \{1, 2\}; d \mapsto \{1\}\}$  satisfait la condition nécessaire indiquée dans le théorème 14. Informellement, nous pouvons bien extraire un sous-graphe acyclique du GCL de la figure 1 où la racine est  $a_3$ , les feuilles sont des solutions vides, et où tous les enfants des nœuds de type solution ou état local sont présents, et au moins un enfant par nœuds de type objectif (figure 2).

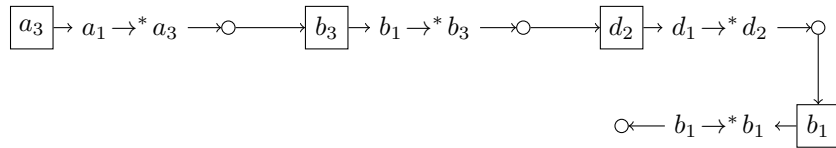


Figure 2. Sous-graphe acyclique du Graphe de Causalité Locale expliquant la satisfaction de la condition nécessaire pour l'accessibilité de  $a_3$  dans le réseau d'automate de l'exemple 6

Dans le même réseau d'automates, l'accessibilité de  $a_3$  depuis le contexte  $\zeta = \{a \mapsto \{2\}; b \mapsto \{1\}; c \mapsto \{1\}; d \mapsto \{2\}\}$  est impossible, car le GCL correspondant (figure 3) ne satisfait pas la condition nécessaire du théorème 14.

□

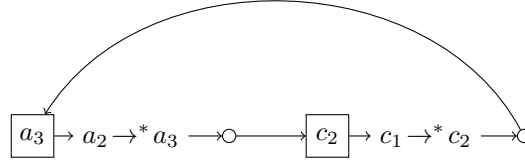


Figure 3. Graphe de Causalité Locale pour l'accessibilité de  $a_3$  dans le réseau d'automate de l'exemple 6 depuis le contexte  $\{a \mapsto \{2\}; b \mapsto \{1\}; c \mapsto \{1\}; d \mapsto \{1\}\}$ .

### 3.3. Application : ensembles de coupe pour l'accessibilité

Si  $a_j$  est accessible depuis un état du contexte  $\varsigma$ , nous pouvons vouloir modifier le réseau d'automates pour rendre cette accessibilité impossible. Ce type de question fait écho à la recherche de cibles (knock-out/knock-in) dans les réseaux biologiques pour contrôler l'émergence d'un phénotype.

Nous appelons ensemble de coupe (ou *cut set*) pour l'accessibilité de  $a_j$  depuis  $\varsigma$  un ensemble d'états locaux  $cs \in \wp(\mathbf{LS})$  du réseau d'automates tel que toute séquence de transitions permettant d'atteindre  $a_j$  contient une transition  $\ell$  avec  $\bullet\ell \cap cs \neq \emptyset$ . Ainsi, désactiver tous les états locaux de  $cs$  du réseau d'automates rend impossible l'accessibilité de  $a_j$ .

Nous montrons ici que le GCL présenté dans cette section permet d'extraire des ensembles de coupe dont la désactivation dans le réseau d'automates empêche la satisfaction de la condition nécessaire du théorème 14. La définition 16 décrit une valuation  $\mathbb{V}$  des ensembles de coupe de chacun des nœuds du GCL : (1) un ensemble de coupe pour une solution est un ensemble de coupe d'un état local la composant ; (2) un ensemble de coupe pour un objectif est un ensemble de coupe pour toutes ses solutions ; (3) un ensemble de coupe pour un état local est un ensemble de coupe pour tous ses objectifs. Seuls les ensembles de coupe minimaux sont considérés.

DÉFINITION 16 (Valuation  $\mathbb{V}$ ). — Une valuation  $\mathbb{V} : V_\varsigma^\omega \rightarrow \wp(\mathbf{LS})$  des nœuds du GCL  $(V_\varsigma^\omega, E_\varsigma^\omega)$  satisfait pour tout  $n \in V_\varsigma^\omega$  :

1.  $n \in \mathbf{Sol} \Rightarrow \mathbb{V}(n) = \zeta(\bigcup_{m \in \text{children}(n)} \mathbb{V}(m))$
2.  $n \in \mathbf{Obj} \Rightarrow \mathbb{V}(n) = \zeta(\tilde{\prod}_{m \in \text{children}(n)} \mathbb{V}(m))\}$
3.  $n \in \mathbf{LS} \Rightarrow \mathbb{V}(n) = \zeta(\{\{a_i\}\} \cup \tilde{\prod}_{m \in \text{children}(n)} \mathbb{V}(m))\}$

avec  $\zeta(\{e_1, \dots, e_k\}) \triangleq \{e_i \mid i \in [1; k] \wedge \nexists j \in [1; k], j \neq i, e_j \subset e_i\}$ .

PROPOSITION 17. — Étant donné un Graphe de Causalité Locale  $(V_\varsigma^\omega, E_\varsigma^\omega)$  pour l'accessibilité de  $\omega = a_j \in \mathbf{LS}$  dans le réseau d'automates  $(\Sigma, S, \mathcal{L}, T)$  depuis un contexte  $\varsigma$ , pour tout  $cs \in \mathbb{V}(a_j)$ ,  $cs \neq \{a_j\}$ ,  $a_j$  n'est accessible depuis aucun état  $s$  de  $\varsigma$  dans le réseau d'automates  $(\Sigma, S, \mathcal{L}, T) \ominus cs$ .

Pour un nœud  $n$  donné,  $\mathbb{V}(n)$  dépend de la valeur  $\mathbb{V}$  de chacun de ses enfants. Ainsi, le calcul de  $\mathbb{V}$  peut suivre l'ordre topologique du graphe des composantes connexes correspondant : en commençant par les enfants, et en propageant aux parents. En présence de cycles, la valeur de  $\mathbb{V}$  des nœuds des composantes fortement connexes est mise à jour jusqu'à convergence. Un tel algorithme est détaillé dans (Paulevé *et al.*, 2013), ainsi que la preuve de sa terminaison et de sa correction. La preuve de terminaison repose sur la monotonie de  $\mathbb{V}$  au cours des itérations. La correction se démontre alors en remarquant qu'en partant de  $\mathbb{V}$  associant l'ensemble vide à chaque nœud, la proposition 17 est vérifiée après chaque mise à jour de  $\mathbb{V}$ .

Le GCL référénçant des conditions seulement nécessaires pour l'accessibilité, il est important de noter que des ensembles de coupe peuvent être ignorés, ou non minimaux pour le réseau d'automates sous-jacent.

EXEMPLE 18. — Appliqué au GCL de la figure 1 pour le réseau d'automates  $(\Sigma, S, \mathcal{L}, T)$  de l'exemple 6 à partir du contexte  $\varsigma$ , tableau 1 donne le détail du calcul des ensembles de coupe pour les nœuds du GLC. Le calcul résulte en  $\mathbb{V}(a_3) = \{\{a_3\}, \{b_1\}, \{b_3, c_2\}, \{c_2, d_2\}\}$ . Ainsi,  $a_3$  n'est pas accessible depuis  $\varsigma$  dans les réseaux  $(\Sigma, S, \mathcal{L}, T) \ominus \{b_1\}$ ,  $(\Sigma, S, \mathcal{L}, T) \ominus \{b_3, c_2\}$ , et  $(\Sigma, S, \mathcal{L}, T) \ominus \{c_2, d_2\}$ .  $\square$

Tableau 1. Calcul itératif de  $\mathbb{V}$  sur le GCL de la figure 1 en suivant un ordre topologique du graphe des composantes connexes correspondant.

Nœud	rang	$\mathbb{V}$
$\langle b_1 \rightarrow^* b_1, \emptyset \rangle$	1	$\emptyset$
$b_1 \rightarrow^* b_1$	2	$\emptyset$
$b_1$	3	$\{\{b_1\}\}$
$\langle d_1 \rightarrow^* d_2, \{b_1\} \rangle$	4	$\{\{b_1\}\}$
$d_1 \rightarrow^* d_2$	5	$\{\{b_1\}\}$
$d_2$	6	$\{\{b_1\}, \{d_2\}\}$
$\langle b_1 \rightarrow^* b_3, \{d_2\} \rangle$	7	$\{\{b_1\}, \{d_2\}\}$
$b_1 \rightarrow^* b_3$	8	$\{\{b_1\}, \{d_2\}\}$
$b_3$	9	$\{\{b_1\}, \{b_3\}, \{d_2\}\}$
$\langle a_1 \rightarrow^* a_3, \{b_3\} \rangle$	10	$\{\{b_1\}, \{b_3\}, \{d_2\}\}$
$\langle c_2 \rightarrow^* c_2, \emptyset \rangle$	11	$\emptyset$
$c_2 \rightarrow^* c_2$	12	$\emptyset$
$c_2$	13	$\{\{c_2\}\}$
$\langle a_1 \rightarrow^* a_3, \{b_1, c_2\} \rangle$	13	$\{\{b_1\}, \{c_2\}\}$
$a_1 \rightarrow^* a_3$	13	$\{\{b_1\}, \{b_3, c_2\}, \{c_2, d_2\}\}$
$a_3$	13	$\{\{a_3\}, \{b_1\}, \{b_3, c_2\}, \{c_2, d_2\}\}$
$\langle c_1 \rightarrow^* c_2, \{a_3\} \rangle$	13	$\{\{a_3\}, \{b_1\}, \{b_3, c_2\}, \{c_2, d_2\}\}$

#### 4. Conditions suffisantes pour l'accessibilité dans les réseaux d'automates asynchrones

Nous présentons dans cette section une condition suffisante à l'existence d'une suite de transitions au sein d'un réseau d'automates asynchrone  $(\Sigma, S, \mathcal{L}, T)$  (définition 3) partant d'un état global initial  $s \in S$  et atteignant un état  $s' \in S$  où  $s(a) = j$  pour un automate  $a \in \Sigma$  et état local  $j \in S(a)$  donnés. À l'instar de la section précédente, la notion d'état est généralisée à la notion de contexte, mais nous nous intéressons cette fois aux conditions suffisantes pour l'accessibilité de  $a_j$  depuis tout état global délimité par le contexte – et non d'au moins un état délimité par le contexte, comme dans la section 3.

Les résultats reposent sur une extension du Graphe de Causalité Locale présentée dans la section précédente pour faire apparaître les états partiels imposés par les pré-conditions de transitions impliquant un nombre arbitraire d'automates.

##### 4.1. Graphe de causalité locale simultanée saturé

Pour un objectif  $P = a_i \rightarrow^* a_j \in \mathbf{Obj}$  donné, nous définissons  $\text{nsol}(P) \subset \wp(\wp(\mathbf{LS}))$  la *causalité locale simultanée* de  $P$  (définition 19) : chaque  $nls \in \text{sol}(P)$ , appelé solution n-aire, correspond à un ensemble de pré-conditions, qui sont des ensembles d'états locaux dont la présence simultanée est requise pour pouvoir appliquer la transition à l'état global du réseau.

**DÉFINITION 19** (Causalité locale simultanée  $\text{nsol}$ ). —  $\text{nsol} : \mathbf{Obj} \rightarrow \wp(\wp(\wp(\mathbf{LS})))$  associe à un objectif un ensemble de solutions n-aires, une solution n-aire étant un ensemble d'ensembles d'états locaux. L'ensemble de ces associations est noté  $\mathbf{NSol} \triangleq \{(P, nls) \mid nls \in \text{nsol}(P)\}$ .

L'ensemble  $\text{nsol}(a_i \rightarrow^* a_j)$  est dit suffisant (définition 20) si et seulement si toute solution n-aire le composant correspond à une séquence acyclique complète de transitions locales au sein de l'automate  $a$  entre ses états locaux  $i$  et  $j$ .

**DÉFINITION 20** ( $\text{nsol}$  suffisant). — *Étant donné un réseau d'automates asynchrone  $(\Sigma, S, \mathcal{L}, T)$  où  $a_i \rightarrow^* a_j \in \mathbf{Obj}$ ,  $\text{nsol}(a_i \rightarrow^* a_j)$  est suffisant si et seulement si, pour tout  $nls \in \text{nsol}(a_i \rightarrow^* a_j)$ , il existe une séquence acyclique  $i \xrightarrow{\ell_1} \dots \xrightarrow{\ell_m} j$  de transitions locales dans  $T(a)$  telle que, pour tout  $\ell \in \{\ell_1, \dots, \ell_m\}$ ,  $\{b_j \in \bullet \ell \mid b \neq a\} \in nls$ .*

La définition 21 étend le graphe de causalité locale de la définition 11 avec la notion de causalité simultanée, et en considérant tous les objectifs possibles impliquant les états locaux référencés dans les solutions n-aires. Ce graphe rend également compte de potentielles décompositions d'objectifs : lors de l'accessibilité de  $a_j$  depuis  $a_i$ , si l'état local  $a_k$  ( $k \notin \{i, j\}$ ) peut être requis avant l'atteinte de  $a_j$  ( $a_k$  appartient à la descendance de  $a_i \rightarrow^* a_j$  dans le graphe),  $a_k \rightarrow^* a_j$  est ajouté en tant que fils de l'objectif  $a_i \rightarrow^* a_j$ . Nous appelons un tel graphe un Graphe de Causalité Locale Simultanée Saturé.

DÉFINITION 21 (Graphe de causalité locale simultanée saturé). — *Étant donné un réseau d'automates  $(\Sigma, S, \mathcal{L}, T)$  avec un contexte  $\varsigma$  et un état local  $\omega \in \mathbf{LS}$ , le graphe de causalité locale simultanée saturé (GCLS-sat)  $(\overline{V}_\varsigma^\omega, \overline{E}_\varsigma^\omega)$ , avec  $\overline{V}_\varsigma^\omega \subseteq \mathbf{LS} \cup \mathbf{Obj} \cup \mathbf{NSol} \cup \wp(\mathbf{LS})$  et  $\overline{E}_\varsigma^\omega \subseteq \overline{V}_\varsigma^\omega \times \overline{V}_\varsigma^\omega$ , est la plus petite structure satisfaisant :*

1.  $\omega \in \overline{V}_\varsigma^\omega$
2.  $a_i \in \overline{V}_\varsigma^\omega \cap \mathbf{LS} \Leftrightarrow \{(a_i, a_j \rightarrow^* a_i) \mid a_j \in \varsigma \vee (a_j \neq a_i \wedge a_j \neq \omega \wedge a_j \in \overline{V}_\varsigma^\omega \cap \mathbf{LS})\} \subseteq \overline{E}_\varsigma^\omega$
3.  $P \in \overline{V}_\varsigma^\omega \cap \mathbf{Obj} \Leftrightarrow \{(P, \langle P, nls \rangle) \mid nls \in \mathbf{nsol}(P)\} \subseteq \overline{E}_\varsigma^\omega$
4.  $\langle P, nls \rangle \in \overline{V}_\varsigma^\omega \cap \mathbf{NSol} \Leftrightarrow \{(\langle P, nls \rangle, ls) \mid ls \in nls\} \subseteq \overline{E}_\varsigma^\omega$
5.  $ls \in \overline{V}_\varsigma^\omega \cap \wp(\mathbf{LS}) \Leftrightarrow \{(ls, a_i) \mid a_i \in ls\} \subseteq \overline{E}_\varsigma^\omega$
6.  $a_i \rightarrow^* a_j \in \overline{V}_\varsigma^\omega \cap \mathbf{Obj} \Rightarrow \{(a_i \rightarrow^* a_j, a_k \rightarrow^* a_j) \mid a_k \neq a_j, a_k \in \mathbf{conn}_{(\overline{V}_\varsigma^\omega, \overline{E}_\varsigma^\omega)}(a_i \rightarrow^* a_j, nls), nls \in \mathbf{nsol}(a_i \rightarrow^* a_j)\} \subseteq \overline{E}_\varsigma^\omega$  .

Le calcul de l'ensemble  $\mathbf{nsol}$  est similaire au calcul de  $\mathbf{sol}$  de la section précédente, à la différence que les pré-conditions ne sont plus fusionnées, ceci afin de conserver l'information sur les présences simultanées requises pour le déclenchement des transitions. Au final, la complexité du GCLS-sat est identique à la complexité du GCL de la section précédente,

EXEMPLE 22. — Soit le réseau d'automates  $(\Sigma, S, \mathcal{L}, T)$  suivant :

$$\begin{aligned} \Sigma &\triangleq \{a, b, c\} & \mathcal{L} &\triangleq \{\ell_1, \ell_2, \ell_3, \ell_4\} \\ S(a) &= [1; 2] & T(a) &\triangleq \{1 \xrightarrow{\ell_1} 2, 2 \xrightarrow{\ell_2} 2, 2 \xrightarrow{\ell_3} 2\} \\ S(b) &\triangleq [1; 2] & T(b) &\triangleq \{2 \xrightarrow{\ell_3} 2, 1 \xrightarrow{\ell_2} 2\} \\ S(c) &\triangleq [1; 2] & T(c) &\triangleq \{1 \xrightarrow{\ell_1} 1, 1 \xrightarrow{\ell_3} 2, 2 \xrightarrow{\ell_4} 1\} \end{aligned}$$

$(\Sigma, S, \mathcal{L}, T)$  est un réseau asynchrone. Le graphe de causalité locale simultanée saturé pour l'accessibilité de  $c_2$  depuis le contexte  $\varsigma = \{a \mapsto \{1\}, b \mapsto \{1\}, c \mapsto \{1\}\}$  est donné dans la figure 4. □

#### 4.2. Motif du GCLS-sat suffisant pour l'accessibilité

Le théorème 23 établit une condition suffisante pour qu'un état local  $a_j$  soit accessible depuis n'importe quel état délimité par un contexte donné. La preuve que nous esquissons ici est détaillée dans (Folschette *et al.*, 2013). Cette condition repose sur l'absence de cycle dans le graphe de causalité locale simultanée saturé, et sur l'indépendance de l'accessibilité des états locaux de chaque pré-condition des transitions considérées.



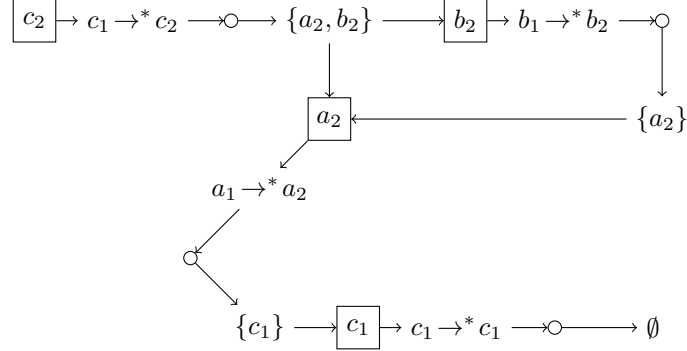


Figure 4. Graphe de Causalité Locale Saturé pour l'accessibilité de  $c_2$  dans le réseau d'automate asynchrone de l'exemple 22

Ce théorème ne s'applique que pour les réseaux d'automates asynchrones. En effet le graphe de causalité locale ne capture que la cause du changement de l'état local d'un automate, et ne tient pas compte des effets de bords potentiels, comme le changement simultané d'états d'autres automates.

**THÉORÈME 23** (Condition suffisante pour l'accessibilité). — Soit  $(\Sigma, S, \mathcal{L}, T)$  un réseau d'automates asynchrone,  $\omega = a_j \in \mathbf{LS}$  un de ses états locaux, et  $\varsigma$  un contexte pour le réseau.  $a_j$  est accessible depuis tout état délimité par  $\varsigma$  si le Graphe de Causalité Locale Saturé  $(\overline{V}_\varsigma^\omega, \overline{E}_\varsigma^\omega)$  satisfait les conditions suivantes :

1.  $(\overline{V}_\varsigma^\omega, \overline{E}_\varsigma^\omega)$  est sans cycle ;
2.  $\forall P \in \overline{V}_\varsigma^\omega \cap \mathbf{Obj}$ ,  $\text{nsol}(P)$  est suffisant et  $\text{nsol}(P) \neq \emptyset$  ;
3.  $\forall ls \in \overline{V}_\varsigma^\omega \cap \wp(\mathbf{LS})$  où  $|ls| \geq 2$ , l'ensemble d'états locaux  $cls = ls \cup \text{conn}_{(\overline{V}_\varsigma^\omega, \overline{E}_\varsigma^\omega)}(ls)$  contient au plus un état local par automate :  $\forall a \in \Sigma, S(a) \cap ls \neq \emptyset \Rightarrow |S(a) \cap cls| \leq 1$ .

**PREUVE 24.** — Par induction : étant donné un nœud  $n$  du graphe  $(\overline{V}_\varsigma^\omega, \overline{E}_\varsigma^\omega)$ , on démontre que si tout nœud de sa descendance  $\text{conn}_{(\overline{V}_\varsigma^\omega, \overline{E}_\varsigma^\omega)}(n)$  est satisfiable,  $n$  est satisfiable depuis  $\varsigma$ , alors  $n$  est satisfiable depuis  $\varsigma$ . Si  $n \in \mathbf{LS}$ ,  $n = a_i$  est dit satisfiable si et seulement si pour tout état  $s \in \varsigma$ , il existe un état  $s' \in S$  où  $s'(a) = j$  et  $s'$  est accessible depuis  $s$ . Si  $n \in \wp(\mathbf{LS})$ ,  $n = ls$  est dit satisfiable si et seulement pour tout état  $s \in \varsigma$ , il existe un état  $s' \in S$  où  $\forall b_k \in ls, s'(b) = k$  et  $s'$  est accessible depuis  $s$ . Si  $n \in \mathbf{NSol}$ ,  $n = \langle a_i \rightarrow^* a_j, \{ls^1, \dots, ls^x\} \rangle$  est dit satisfiable si pour toute permutation  $\sigma : [1; x] \rightarrow [1, x]$ , pour tout  $s \in \varsigma$ , pour  $y \in [2; x]$ ,  $ls^{\sigma(y)}$  est satisfiable depuis l'état où  $ls^{\sigma(y-1)}$  est satisfait.

Si  $n \in \mathbf{Obj}$ ,  $n = a_i \rightarrow^* a_j$ , si il existe  $k \in S(a)$  tel que  $a_k \rightarrow^* a_j \in \text{conn}_{(\overline{V}_\varsigma^\omega, \overline{E}_\varsigma^\omega)}(n)$ , alors par construction de  $(\overline{V}_\varsigma^\omega, \overline{E}_\varsigma^\omega)$  et par induction,  $n$  est satisfiable. Si il n'existe pas un tel  $k$ , d'après la condition (1) il existe au moins une solution  $\langle a_i \rightarrow^* a_j, cls \rangle$  enfant de  $n$ . Par hypothèse cette solution est satisfiable, ce qui im-

plique qu'il existe une séquence d'états  $s^1, \dots, s^x$  pour lequel il existe une séquence acylique de transitions locales  $i \xrightarrow{\ell^1} \dots \xrightarrow{\ell^x} j$  dans  $T(a)$  où, pour tout  $y \in [1; x]$ ,  $\bullet \ell^y \subset s^1$  et où  $s^y$  est accessible depuis  $s^{y-1}$ , pour tout  $s^0 \in \varsigma$ .

Si  $n \in \mathbf{LS}$ ,  $n = a_j$  est satisfiable depuis un état  $s$  si l'objectif  $s(a) \rightarrow^* a_j$  est un de ses enfants. Par construction de  $(\overline{V}_\varsigma^\omega, \overline{E}_\varsigma^\omega)$ , ceci est toujours vrai.

Si  $n \in \wp(\mathbf{LS})$ , par la condition (3), si  $n = \{\omega^1, \dots, \omega^x\}$ , pour tout  $y, z \in [1; x]$ , si  $s^y$  est un état accessible depuis  $s \in \varsigma$  où  $\omega^y \in s^y$ , il existe un état  $s^z$  accessible depuis  $s^y$  où  $\omega^y \in s^z$  et  $\omega^z \in s^z$ . Ainsi il existe toujours un état  $s'$  accessible depuis tout état  $s \in \varsigma$  tel que,  $\forall y \in [1; x]$ ,  $\omega^y \in s'$ .

Enfin, le graphe  $(\overline{V}_\varsigma^\omega, \overline{E}_\varsigma^\omega)$  étant sans cycle, par construction et par la condition (2), le raisonnement par induction se termine toujours sur la feuille  $\emptyset \in \wp(\mathbf{LS})$ , qui est satisfiable depuis  $\varsigma$ . ■

EXEMPLE 25. — D'après le graphe de causalité locale simultanée saturé de l'exemple 22, le théorème 23 permet de conclure que l'accessibilité de  $c_2$  est possible depuis l'état  $\langle 1, 1, 1 \rangle$  (ordre  $a, b, c$ ). À partir de la preuve du théorème 23, nous pouvons extraire la suite de transition permettant son accessibilité :  $\ell_1, \ell_2, \ell_3$ , donnant la suite d'états  $\langle 1, 1, 1 \rangle \rightarrow \langle 2, 1, 1 \rangle \rightarrow \langle 2, 2, 1 \rangle \rightarrow \langle 2, 2, 2 \rangle$ . □

## 5. Points fixes dans les Frappes de Processus

Dans cette section, nous nous intéressons à la caractérisation des points fixes d'un réseau d'automates (définition 26).

DÉFINITION 26 (Point fixe d'un réseau d'automates). — *Étant donné un réseau d'automates  $\mathcal{N} = (\Sigma, S, \mathcal{L}, T)$ , l'état  $s \in S$  est un point fixe si et seulement si il n'existe aucun  $s' \neq s$  tel que  $s \rightarrow s'$ .*

Nous nous focalisons sur les réseaux d'automates asynchrones de type Frappes de Processus (définition 4), c.-à-d. où le cardinal des pré-conditions des transitions est au plus deux. Nous notons toutefois que tout réseau d'automates, réseau discret, ou réseau de Petri borné peut se traduire en Frappes de Processus en conservant les points fixes (Folschette *et al.*, 2013 ; Paulevé *et al.*, 2012 ; Paulevé, 2011).

Le calcul efficace des points fixes d'un réseau discret a été notamment abordé par Naldi *et al.* (2007) avec l'utilisation de diagramme de décision pour encoder les fonctions déterministes régissant la dynamique. Le résultat que nous présentons en section 5.1 donne une méthode alternative, et permet de mettre en avant le lien très fort entre la topologie du réseau d'automates et ses points fixes : les points fixes sont exactement les cliques composées de  $n$  nœuds, où  $n$  est le nombre d'automates, dans une représentation graphique particulière des Frappes de Processus.

### 5.1. Condition nécessaire et suffisante

Étant données des Frappes de Processus  $(\Sigma, S, \mathcal{L}, T)$ , le Graphe Sans-Frappe (définition 27) met en relation les états locaux du réseau qui ne partagent pas de libellé de transitions locales.

**DÉFINITION 27** (Graphe Sans-Frappe). — *Le Graphe Sans-Frappe des Frappes de Processus  $(\Sigma, S, \mathcal{L}, T)$  est un graphe  $(V, E)$  non-orienté où les sommets  $V$  et les arcs  $E$  sont définis de la manière suivante :*

$$V \triangleq \{a_i \in \mathbf{LS} \mid \nexists \ell \in \mathcal{L}, \bullet \ell = \{a_i\}\} ,$$

$$E \triangleq \{\{a_i, b_j\} \subseteq V \mid a \neq b \wedge \nexists \ell \in \mathcal{L}, \bullet \ell = \{a_i, b_j\}\} .$$

Le théorème 28 caractérise complètement les points fixes des Frappes de Processus  $(\Sigma, S, \mathcal{L}, T)$  à partir du Graphe Sans-Frappe.

**THÉORÈME 28.** — *Étant donné des Frappes de Processus  $(\Sigma, S, \mathcal{L}, T)$ , un état  $s \in S$  est un point fixe si et seulement si  $s$  est une clique composée de  $|\Sigma|$  nœuds du Graphe Sans-Frappe  $(V, E)$  associé.*

**PREUVE 29.** —  $s$  est un point fixe pour  $(\Sigma, S, \mathcal{L}, T) \iff \forall a, b \in \Sigma (\nexists \ell \in \mathcal{L}, \bullet \ell = \{a_{s(a)}, b_{s(b)}\} \iff \{a_{s(a)}, b_{s(b)}\} \in E)$ . ■

Nous remarquons que le Graphe Sans-Frappe étant  $|\Sigma|$ -parti, la recherche de cliques composées de  $|\Sigma|$  sommets peut se décrire directement en SAT pour lesquels des implémentations efficaces existent, bien que de complexité exponentielle au nombre d'automates (le problème de recherche de points fixes est NP-complet).

**EXEMPLE 30.** — La figure 5 illustre le théorème 28 : le réseau ne possède qu'un seul point fixe correspondant à l'unique clique de trois sommets entre  $a_0, b_0, c_0$  (lignes épaisses). □

## 6. Conclusion

Les réseaux d'automates constituent un paradigme de modélisation adapté pour capturer la complexité des systèmes concurrents. Ils permettent de représenter l'évolution de différents agents en interaction. En biologie, notamment, ils se sont imposés comme l'un des formalismes les plus appréciés. Dans ce contexte, un des enjeux de la modélisation est d'affiner la connaissance des influences locales à partir d'observations faites sur l'ensemble du système.

Cependant, la connaissance des systèmes réels est souvent entachée d'incertitudes (liées par exemple à une difficulté d'expérimentation pour mesurer telle ou telle concentration). Il est de fait nécessaire de ne pas trop contraindre le modèle *a priori*, et de procéder plutôt par raffinements successifs. Qui plus est, la complexité de ces systèmes rend impossible l'analyse de leur dynamique en se basant sur une exploration exhaustive de leur espace d'états. Pour contourner cette difficulté, il est crucial

$$\begin{aligned} \Sigma &= \{a, b, c\} \\ S(a) &= [0; 1] \quad S(b) = [0; 1] \\ S(c) &= [0; 1] \\ \mathcal{L} &= \{\ell_1, \ell_2, \ell_3, \ell_4, \ell_5\} \\ T(a) &= \{0 \xrightarrow{\ell_5} 1, 1 \xrightarrow{\ell_2} 0, 1 \xrightarrow{\ell_4} 0\} \\ T(b) &= \{0 \xrightarrow{\ell_4} 0, 1 \xrightarrow{\ell_5} 1, 1 \xrightarrow{\ell_3} 3\} \\ T(c) &= \{0 \xrightarrow{\ell_3} 0, 1 \xrightarrow{\ell_1} 0, 1 \xrightarrow{\ell_2} 1\} \end{aligned}$$

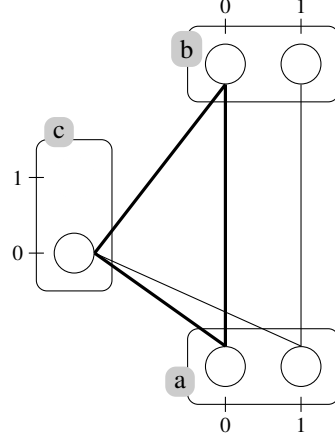


Figure 5. Frappes de Processus (gauche) et leur Graphe Sans-Frappe (droite)

de concevoir des méthodes statiques permettant une étude efficace des propriétés du modèle. Ce sont ces deux idées clefs qui ont gouverné le développement de l'approche de modélisation et d'analyse présentée dans ce papier.

Les résultats résumés dans cet article s'appliquent dans le cadre des réseaux d'automates indéterministes, où l'évolution de l'état des composants est défini par des transitions locales dont les conditions d'applications ne sont pas nécessairement disjointes. Cette vision centrée sur les transitions met en exergue la concurrence entre les mises à jour des automates d'un réseau, et est exploitée pour réduire l'ensemble des comportements à considérer en ignorant, notamment, de nombreux entrelacements. Notre approche suit un cadre typique d'interprétation abstraite : à partir du modèle concret, nous extrayons un ensemble d'informations que nous résumons dans un graphe dit de causalité locale. Nous établissons alors des motifs au sein de ces graphes qui sont soit nécessaires, soit suffisants, à l'accessibilité de l'état local d'un automate donné. L'avantage de cette approche est sa complexité très réduite – exponentielle selon le nombre d'états locaux au sein d'un seul automate ; polynomiale selon le nombre d'automates et de transitions locales – comparée à la complexité du problème de l'accessibilité qui est PSPACE complet. Bien sûr, nos conditions étant soit nécessaires, soit suffisantes, les théorèmes présentés ici peuvent s'avérer non concluants pour certains réseaux ou propriétés.

L'aspect indéterministe des réseaux étudiés dans cet article, outre leur généralisation des réseaux d'automates classiques où la dynamique locale des composants est déterministe, apporte un outil pratique pour la modélisation des réseaux biologiques avec une connaissance partielle sur les fonctions d'évolution. En effet, comme détaillé dans (Paulevé *et al.*, 2011 ; Paulevé, 2011 ; Folschette *et al.*, 2012), étant donné un graphe d'influence, et sous l'hypothèse de monotonie des fonctions d'évolution de chaque composant, il est possible de construire de manière compacte un réseau d'automates indéterministe dont la dynamique est l'union des transitions de tous les

réseaux discrets compatibles avec le graphe d'influence. Il est également possible de construire des réseaux d'automates incluant la dynamique d'un certain ensemble de réseaux discrets soumis à certaines contraintes sur les fonctions d'évolution. Ainsi, les réseaux d'automates indéterministes offrent un cadre simple pour raisonner avec un seul modèle dynamique sur un ensemble, potentiellement très grand, de modèles dynamiques. Notamment, si une propriété d'accessibilité est démontrée impossible, elle est impossible dans tout réseau ayant sa dynamique incluse. Les ensembles de coupes calculés sur un réseau sont également valables pour tout réseau ayant sa dynamique incluse dans le premier. Et il en va de même pour les points fixes : tout point fixe d'un réseau d'automates est un point fixe pour les réseaux d'automates avec une dynamique plus restreinte. Cette dernière remarque, appliquée aux réseaux d'automates indéterministes incluant la dynamique de tout réseau discret partageant le même graphe d'influence, permet de retrouver, via le théorème 28, la caractérisation des points fixes topologiques (Paulevé, Richard, 2010) des réseaux booléens.

Comme détaillé dans (Paulevé *et al.*, 2012 ; Folschette *et al.*, 2013 ; Paulevé *et al.*, 2013), ces analyses statiques ont été appliquées pour l'analyse de la dynamique de très grands réseaux de régulation et de signalisation. Pour des réseaux contenant des centaines de composants, les outils classiques de vérification formelles ne permettent pas de répondre en un temps et espace mémoire raisonnables ; nos analyses, de par leur faible complexité, répondent en quelques centièmes de seconde. Pour les réseaux particuliers étudiés, il s'est trouvé que nos analyses ont toujours été concluantes. Le calcul des ensembles de coupes pour le contrôle de l'accessibilité a été appliqué à un réseau de 9000 composants, représentant une base de donnée complète d'interactions. C'est à notre connaissance la seule application de méthodes formelles d'analyse dynamique à des réseaux d'une telle taille à ce jour.

Maintenant que nous avons conçu une approche efficace pour l'analyse d'accessibilité, nous souhaitons porter nos efforts sur d'autres propriétés importantes pour de nombreux systèmes. Et, en premier lieu, nous souhaitons exhiber des motifs nécessaires ou suffisants pour la présence d'attracteurs cycliques. Nous prévoyons également d'approfondir le lien entre un changement d'un élément de la structure du modèle, et l'impact de ce changement sur les propriétés dynamiques du modèle. Ce travail devrait rejoindre les recherches liant topologie d'un réseau d'interaction avec ses caractéristiques dynamiques.

## Bibliographie

- Aracena J., Goles E., Moreira A., Salinas L. (2009). On the robustness of update schedules in boolean networks. *Biosystems*, vol. 97, n° 1, p. 1 - 8.
- Baldan P., Chatain Th., Haar S., König B. (2010). Unfolding-based diagnosis of systems with an evolving topology. *Information and Computation*, vol. 208, n° 10, p. 1169-1192.
- Bernardinello L., De Cindio F. (1992). A survey of basic net models and modular net classes. In G. Rozenberg (Ed.), *Advances in petri nets 1992*, vol. 609, p. 304-351. Springer Berlin / Heidelberg.

- Chaouiya C., Naldi A., Remy E., Thieffry D. (2011). Petri net representation of multi-valued logical regulatory graphs. *Natural Computing*, vol. 10, n° 2, p. 727-750.
- Cinquin O., Demongeot J. (2002). Roles of positive and negative feedback in biological systems. *Comptes Rendus Biologies*, vol. 325, n° 11, p. 1085 - 1095.
- Comet J.-P., Noual M., Richard A., Aracena J., Calzone L., Demongeot J. *et al.* (2013). On circuit functionality in boolean networks. *Bulletin of Mathematical Biology*, vol. 75, n° 6, p. 906-919.
- Folschette M., Paulevé L., Inoue K., Magnin M., Roux O. (2012). Concretizing the process hitting into biological regulatory networks. In D. Gilbert, M. Heiner (Eds.), *Computational methods in systems biology*, p. 166-186. Springer Berlin Heidelberg.
- Folschette M., Paulevé L., Magnin M., Roux O. (2013). Under-approximation of reachability in multivalued asynchronous networks. *Electronic Notes in Theoretical Computer Science*, vol. 299, p. 33 - 51. (4th International Workshop on Interactions between Computer Science and Biology (CS2Bio'13))
- Naldi A., Thieffry D., Chaouiya C. (2007). Decision diagrams for the representation and analysis of logical models of genetic networks. In M. Calder, S. Gilmore (Eds.), *Computational methods in systems biology*, vol. 4695, p. 233-247. Springer Berlin Heidelberg.
- Noual M., Regnault D., Sené S. (2012). About non-monotony in Boolean automata networks. *Theoretical Computer Science*, vol. 504, p. 12-25.
- Paulevé L. (2011). *Modélisation, simulation et vérification des grands réseaux de régulation biologique*. PhD thesis, École Centrale de Nantes.
- Paulevé L., Andrieux G., Koepl H. (2013). Under-approximating cut sets for reachability in large scale automata networks. In N. Sharygina, H. Veith (Eds.), *Computer aided verification*, vol. 8044, p. 69-84. Springer Berlin Heidelberg.
- Paulevé L., Magnin M., Roux O. (2011). Refining dynamics of gene regulatory networks in a stochastic  $\pi$ -calculus framework. In *Transactions on computational systems biology xiii*, vol. 6575, p. 171-191. Springer.
- Paulevé L., Magnin M., Roux O. (2012). *From the Process Hitting to Petri Nets and Back*. Technical Report n° hal-00744807.
- Paulevé L., Magnin M., Roux O. (2012). Static analysis of biological regulatory networks dynamics using abstract interpretation. *Mathematical Structures in Computer Science*, vol. 22, n° 04, p. 651-685.
- Paulevé L., Richard A. (2010). Topological Fixed Points in Boolean Networks. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, vol. 348, n° 15-16, p. 825 - 828.
- Remy É., Ruet P., Thieffry D. (2008). Graphic requirements for multistability and attractive cycles in a boolean dynamical framework. *Advances in Applied Mathematics*, vol. 41, n° 3, p. 335 - 350.
- Richard A. (2010). Negative circuits and sustained oscillations in asynchronous automata networks. *Advances in Applied Mathematics*, vol. 44, n° 4, p. 378 - 392.
- Thomas R. (1973). Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, vol. 42, n° 3, p. 563 - 585.

Thomas R. (1981). On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations. In J. Della Dora, J. Demongeot, B. Lacolle (Eds.), *Numerical methods in the study of critical phenomena*, vol. 9, p. 180-193. Springer Berlin Heidelberg.