Les réseaux d'automates booléens au cœur du calcul naturel

Kévin Perrot [1] et Sylvain Sené [2]

Depuis la naissance de la science informatique moderne dans les années 1930-1940, la biologie n'a eu de cesse d'inspirer ses développements sur les plans théorique et appliqué (réseaux de neurones, automates cellulaires, calcul moléculaire...). Réciproquement, depuis les années 1960, l'informatique joue un rôle fondamental dans les avancées en sciences du vivant. Il n'y a qu'à voir les innovations réalisées en algorithmique des séquences, en biologie structurale et en phylogénie pour se rendre compte de l'influence de notre science sur la biologie. Au delà de ces champs de recherche, Jacob et Monod [9] ont mis en avant le caractère essentiel des régulations biologiques, et donc de la transmission d'information. Leurs travaux ont ouvert la voie à un nouveau domaine, ancré en mathématiques et en informatique et visant la compréhension du vivant, s'attachant à analyser la manière dont les entités biologiques interagissent pour engendrer les phénomènes observés. Du point de vue de l'informatique, il s'agit de comprendre comment ces transmissions d'information engendrent les calculs opérés par les organismes étudiés, ce qui correspond exactement à la définition originelle de *bio-informatique* telle qu'introduite par Hesper et Hogeweg [8] : « *l'étude des processus informatiques dans les systèmes biologiques* ».

Dans ce cadre, nous proposons un survol introductif du modèle des réseaux d'automates booléens. Après avoir défini l'objet mathématique, nous parcourrons son histoire et les liens qu'il entretient avec l'informatique et la biologie. Une première série de résultats théoriques montre que les rétroactions jouent un rôle moteur dans

1. Université d'Aix-Marseille, CNRS, LIS, Marseille.
2. Université publique, Marseille.

DOI:10.48556/SIF.1024.20.171 **1024** – Bulletin de la SiF, numéro 20, Novembre 2022, pp. 171–182

la complexité des phénomènes de régulation, et une seconde série témoigne de la difficulté à analyser la dynamique de tels systèmes.

Les réseaux d'automates booléens

Le modèle mathématique des réseaux d'automates capture de façon générale les systèmes composés d'entités en interaction (qui incluent donc les phénomènes de régulation). Un réseau d'automates peut être vu comme un ensemble de machines exécutant des calculs (les automates), qui dépendent des informations qu'elles échangent avec d'autres machines du réseau. Dans le contexte de cet article, à des fins de simplification, nous nous restreignons aux réseaux d'automates booléens, dans lesquels les informations transmises et retenues par les automates, sont binaires (0 ou 1). Plus formellement, dans un *réseau d'automates booléens* de taille n (composé de n automates indexés par $[n] = \{0, \dots, n-1\}$), chaque *automate* $i \in [n]$ possède un *état booléen* $x_i \in \{0,1\}$, et met à jour son état selon une *fonction locale* $f_i : \{0,1\}^n \to \{0,1\}$. L'affectation d'un état à tous les automates du réseau donne une *configuration* $x \in \{0,1\}^n$. Les fonctions locales définissent les influences des automates les uns sur les autres, capturées par un *graphe d'interaction*. Il s'agit d'un graphe orienté composé d'un sommet par automate, avec un arc de l'automate i vers l'automate j lorsque l'actualisation de l'état de j, c'est-à-dire $f_j(x)$, dépend effectivement de l'état de i, c'est-à-dire x_i. On ajoute généralement des signes à ces dépendances, pour spécifier la nature de l'influence : un signe $+$ sur l'arc (i,j) spécifie que l'automate j tend à mimer l'état de l'automate i ; un signe $-$ souligne quant à lui que j tend à prendre l'état opposé de celui de i. Par exemple, pour la fonction $f_j(x) = x_i \vee \neg x_j$, le graphe d'interaction correspondant a un arc positif (i,j) et une boucle négative (j,j). Dans le cas d'une fonction $f_j(x) = x_i \oplus x_j$ (avec $\oplus$ le *ou exclusif*), le graphe d'interaction contient l'arc (i,j) et la boucle (j,j) signés à la fois positivement et négativement.

La donnée des fonctions locales est une description statique (syntaxique) du réseau. Pour obtenir un système dynamique discret sur l'espace $\{0,1\}^n$ qui permettrait d'appréhender la transmission d'information et les calculs opérés, il faut en outre lui associer un *mode de mise à jour*. Ce dernier détermine la manière dont les automates mettent à jour leur état dans le temps discret. Parmi les modes de mise à jour les plus présents dans la littérature, se trouvent :

— le mode *parallèle*, qui est déterministe et qui actualise l'état de chaque automate à chaque étape de temps ;

— le mode parfaitement *asynchrone*, qui est non-déterministe et qui actualise l'état d'un automate à chaque étape de temps.

Le couple (f, μ), avec f un réseau d'automates et μ un mode de mise à jour, définit un système dynamique $f_\mu : \{0,1\}^n \to \{0,1\}^n$, qui peut être représenté par un *graphe de transition*. Cette description dynamique (sémantique) est un graphe orienté qui a

pour ensemble de sommets $\{0,1\}^n$ (l'espace des configurations), et un arc de x vers y, appelé transition, lorsque $f_\mu(x) = y$ dans le cas déterministe, ou $y \in f_\mu(x)$ dans le cas non-déterministe. On a par exemple :

— $f_{\mathsf{p}}(x) = (f_0(x), \ldots, f_{n-1}(x))$ pour le mode parallèle p ;
— $f_{\mathsf{a}}(x) = \{(x_0, \ldots, x_{i-1}, f_i(x), x_{i+1}, \ldots, x_{n-1}) \mid i \in [n]\}$ pour le mode parfaitement asynchrone a.

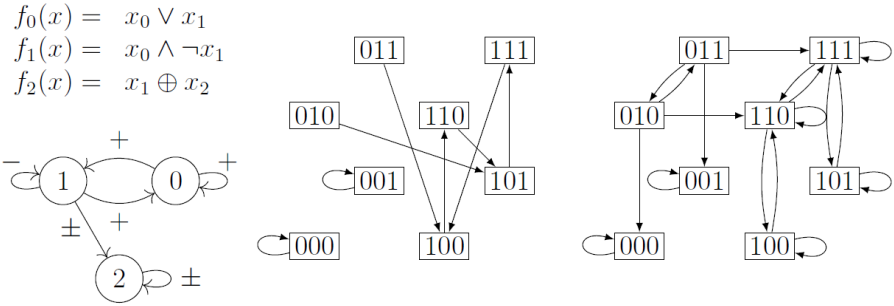

FIGURE 1. Fonctions locales d'un réseau d'automates booléens de taille $n = 3$ (à gauche en haut), son graphe d'interaction (à gauche en bas, avec $\pm$ pour les dépendances à la fois positives et négatives), son graphe de transition pour le mode de mise à jour parallèle (au centre), et son graphe de transition pour le mode de mise à jour parfaitement asynchrone (à droite). Dans le mode parallèle la configuration 011 a pour image $f_{\mathsf{p}}(011) = 100$ où tous les automates changent d'état (en effet, $f_0(011) = 0 \vee 1 = 1$, $f_1(011) = 0 \wedge \neg 1 = 0$ et $f_2(011) = 1 \oplus 1 = 0$), ce qui se traduit dans le mode parfaitement asynchrone par trois transitions possibles $f_{\mathsf{a}}(011) = \{111, 001, 010\}$ où chacun des automates peut mettre à jour son état de façon non-déterministe.

Un exemple est proposé en figure 1. Sur cette figure, on observe que les dynamiques parallèle et asynchrone sont bien différentes, ce qui souligne la sensibilité des réseaux aux modes de mise à jour. Dans la dynamique parallèle, chaque configuration a une et une seule transition possible[3] alors que, dans le cas asynchrone, chaque configuration peut avoir plusieurs transitions possibles, comme par exemple la configuration 010. Ce qui nous intéresse en termes de transmission d'information et de calcul, ce sont les chemins dans la dynamique (trajectoires), et ce vers

3. Cette propriété est vérifiée par tout mode de mise à jour déterministe.

quoi ces chemins mènent (comportements limites, qui sont les composantes fortement connexes terminales[4] du graphe de transition). Deux types de comportement limite nous intéressent particulièrement : les configurations stables et les oscillations stables, communément appelés points fixes et cycles limites dans le cas déterministe. Dans l'exemple de la figure 1, on observe que la dynamique selon le mode parallèle possède deux points fixes $f_{\mathsf{p}}(000) = 000$ et $f_{\mathsf{p}}(001) = 001$, et un cycle limite de période quatre $f_{\mathsf{p}}^4(100) = f_{\mathsf{p}}^3(110) = f_{\mathsf{p}}^2(101) = f_{\mathsf{p}}(111) = 100$. On remarque également que les points fixes du mode parallèle sont aussi des points fixes du mode asynchrone, car aucun automate ne change d'état lors de l'application de sa fonction locale. Dans le mode asynchrone, on a de plus une oscillation stable composée des configurations $100, 110, 101, 111$ (il s'agit ici du même ensemble de configurations que le cycle limite du mode parallèle, mais ce n'est pas toujours le cas). Une trajectoire est une séquence de configurations obtenues par des transitions successives dans le système dynamique, par exemple $(010, 011, 111, 111, 101, 111, 110)$ pour le mode asynchrone sur la figure 1.

Un modèle de calcul naturel

Du point de vue de l'informatique et des mathématiques, les réseaux d'automates sont une famille de systèmes dynamiques discrets, qui peut être utilisée pour capturer des phénomènes biologiques. Ces phénomènes biologiques sont en général décrits à la suite d'expérimentations en laboratoire « humide ». Cette première description peut être assez éloignée du formalisme des réseaux d'automates booléens (même si, en pratique, le langage utilisé est proche de la logique propositionnelle), ce qui nécessite une étape de *traduction*. Par exemple, dans le cadre de la modélisation des réseaux de régulation génétique, chaque gène est représenté par un automate (pour faire simple) dont l'état booléen représente son état d'expression (1 pour activé, 0 pour inhibé). Cet état représente le fait qu'il soit transcrit (ou non) en ARN messager puis traduit (ou non) en protéines. L'expression d'un gène dépend essentiellement de l'expression des autres gènes de son réseau (plus précisément des taux de concentration des protéines produites). On peut alors décrire des observations sous la forme « si les gènes A et B sont exprimés, alors le gène C est inhibé ». Lorsque de telles spécifications sont incomplètes (ce qui est toujours le cas), on peut construire plusieurs réseaux d'automates qui les satisfont. Des informations supplémentaires (observations temporelles de rapidité des réactions en chaîne, différents paliers d'activation...) peuvent permettre de réduire cet ensemble de modèles. On a alors deux types de *modèle* : d'un côté le modèle mathématique des réseaux d'automates, de l'autre des « modèles biologiques » correspondant à des instances du modèle mathématique. L'intérêt de l'analyse de la dynamique limite de tels modèles réside dans le fait que les configurations et les oscillations stables du réseau d'automates sont

4. C'est-à-dire desquelles ne sort aucune transition.

assimilables aux phénotypes (types cellulaires, tissus...) et aux rythmes biologiques (circadien, cardio-respiratoire, cycle cellulaire...) émergeant du système.

Ce qui est certain, c'est qu'aucun modèle biologique ne pourra représenter parfaitement la réalité. En effet, étant une abstraction d'observations incomplètes de la réalité, on ne peut espérer que ces modèles capturent la totalité des paramètres en jeu dans le système réel. Comme l'écrivent Delahaye et Rechenmann [5] : « *Tout modèle est faux, et c'est très bien !* ». La qualité d'un modèle s'évalue à son utilité plutôt qu'à sa correction au regard de la réalité. Les modèles des réseaux d'automates en sont de parfaits exemples. La simplicité de leur définition force leur incorrection vis-à-vis des systèmes qu'ils modélisent, de même qu'elle est grandement utile car elle fait porter l'attention sur l'essence des interactions et des transmissions d'information qui engendrent la complexité observée. Il s'agit ainsi d'une modélisation qualitative permettant d'énoncer des propriétés générales des systèmes, car cette simplicité « libère » les modèles de nombreux paramètres.

Jusque là, nous avons présenté le modèle des réseaux d'automates au travers d'allers-retours entre la biologie et l'informatique mais, une fois passé au modèle mathématique, il convient de l'étudier en profondeur *per se*. Ces inspirations tirées de phénomènes réels offrent en effet de nouveaux objets aux mathématiciens et informaticiens, posant de nouvelles questions sur la théorie des systèmes dynamiques, la combinatoire, et la science du calcul (en particulier au travers de la complexité et de la calculabilité) : un réseau d'automates associé à un mode de mise à jour peut être vu comme un programme, dont la dynamique est l'exécution d'un calcul. S'inspirer de la phénoménologie, en tirer des modèles de calcul, les étudier en profondeur, puis les utiliser pour mieux comprendre les phénomènes naturels, est ce qui fonde le domaine du *calcul naturel*. D'un côté, il s'agit d'envisager la nature comme un ensemble de processus calculatoires à différentes échelles, et d'utiliser la science informatique pour les analyser ; de l'autre, il s'agit d'observer la nature pour découvrir et imaginer de nouvelles façons de calculer. Cette vision du calcul au sein de la nature, et des aspects naturels du calcul, soulève un intérêt croissant de la communauté scientifique, et nous croyons qu'il s'agit d'une direction prometteuse à la fois pour mieux comprendre le vivant et pour le développement de nouvelles technologies.

Une histoire croisée

Bien que le calcul naturel paraisse être un domaine assez récent, il existait déjà aux fondements de l'informatique moderne. Outre le modèle des machines de Turing inspiré du mathématicien devant sa feuille [24], les réseaux de neurones ont été introduits par McCulloch et Pitts [12] pour modéliser les processus cérébraux, et les automates cellulaires ont été développés par von Neumann et Ulam [14] pour conceptualiser l'auto-réplication observée en biologie. Par ailleurs, ces trois modèles

ont donné naissance à des pans entiers de l'informatique moderne et à leur développement. Les machines de Turing ont directement permis de définir le concept de calculabilité qui est le premier pilier de la science informatique. Les réseaux de neurones de McCulloch et Pitts ont quant à eux notamment donné naissance à la théorie des automates finis [11], et ont participé à celle de l'intelligence artificielle [21, 13]. En souhaitant « *comparer les automates naturels et artificiels* » et « *abstraire la structure logique de la vie* », von Neumann a créé un des premiers modèles mathématiques d'architecture d'ordinateur parallèle. Ces deux derniers modèles sont en fait des cas particuliers de réseaux d'automates (le premier avec des fonctions locales à seuil, le second sur une grille avec une homogénéité spatiale).

À l'instar de l'informatique moderne qui s'est largement inspirée de la biologie, cette dernière fait largement appel à la première depuis que Jacob et Monod ont défendu le choix d'une approche systémique en science du vivant. Les réseaux d'automates sont notamment devenus un modèle central pour la modélisation qualitative en biologie moléculaire depuis la fin des années 1960. Ils possèdent en effet les bonnes propriétés pour capturer les caractéristiques structurelles et dynamiques des systèmes biologiques au niveau de la cellule, comme les réseaux de régulation génétique. Par ailleurs, la compréhension des processus de régulation est un problème abstrait que la biologie ne peut résoudre par elle-même. Cette observation fondamentale a été faite en parallèle par Kauffman et Thomas dans le contexte de la régulation génétique : ils soulignent dans [10, 22], l'inadéquation des techniques expérimentales pour aborder le problème dans son entièreté. En effet, bien qu'elles permettent d'acquérir une fine connaissance des éléments biologiques étudiés (sujette à l'interprétation humaine, aux erreurs d'observation, aux biais statistiques...), pour des raisons de complexité notamment, elles empêchent de mettre ces connaissances en perspective et d'obtenir suffisamment de recul pour comprendre les causalités en jeu au sein des combinaisons de régulations. Sur la base de ce constat, Kauffman et Thomas introduisirent des premiers modèles de réseaux de régulation génétique en se fondant sur les réseaux d'automates booléens. La communauté scientifique de biologie des systèmes a vite compris l'intérêt de ces réseaux, qui sont rapidement devenus le modèle mathématique le plus utilisé et le plus influent dans le domaine de la modélisation qualitative des phénomènes de régulation biologique.

Les liens entre structure et dynamique

La principale force du modèle mathématique des réseaux d'automates booléens est la portée générale des résultats obtenus. En effet, par sa capacité à abstraire de nombreux détails des systèmes réels modélisés, le modèle mathématique offre un large potentiel d'applications. Ces applications peuvent concerner non seulement la vérification de la cohérence de résultats expérimentaux et l'aide au choix des prochaines expériences à mener, mais encore la prédiction du comportement d'un

système. Voici un premier exemple de résultat théorique associé à de potentielles applications.

Théorème 1 ([20])**.** *Si le graphe d'interaction d'un réseau de n automates est acyclique, alors sa dynamique converge en au plus n étapes vers une unique configuration stable.*

Pour le mode parallèle, le théorème 1 conclut que f_{p}^n est une fonction constante, c'est-à-dire qu'il existe une configuration $\tilde{x} \in \{0,1\}^n$ telle que, pour toute configuration x, on a $f_{\mathsf{p}}^n(x) = \tilde{x}$. Dans le mode parfaitement asynchrone, le théorème 1 conclut que f_{a} possède une trajectoire de longueur au plus n depuis toute configuration x vers la configuration $\tilde{x}$. Il peut alors être appliqué pour vérifier la cohérence d'un modèle biologique et aider au choix des expériences à mener : si plusieurs configurations stables sont observées mais que le graphe d'interaction est acyclique, c'est que le modèle est incohérent, et qu'il doit exister des interactions créant des cycles dans le graphe d'interaction. Le théorème 1 permet également de prédire le comportement d'un modèle biologique : si le graphe d'interaction est acyclique et que la dynamique est inconnue, on peut prédire qu'il y aura une unique configuration stable. Les résultats théoriques donnent en outre des pistes d'explication mathématique à des phénomènes observés. Par exemple, le théorème 1 montre que les cycles dans le graphe d'interaction, également appelés *boucles de rétroaction* (où une entité agit, parfois indirectement, sur elle-même), sont des moteurs de la complexité dynamique : sans eux, l'évolution est « simple » car, peu importe la configuration initiale, le système convergera toujours vers la même configuration stable.

Parmi les cycles du graphe d'interaction, on distingue les cycles positifs qui comportent un nombre pair d'arcs négatifs, et les cycles négatifs qui comportent un nombre impair d'arcs négatifs. Sur l'exemple de la figure 1, le cycle entre les automates 0 et 1 est positif (zéro arc négatif), alors que le cycle de l'automate 1 vers lui-même est négatif (un arc négatif). L'état d'un automate est « inversé » lorsque qu'il traverse un arc négatif. En revanche, en traversant un nombre pair d'arcs négatifs, cette information revient « à l'identique » sur l'automate lui-même. Un cycle positif est donc intuitivement « bon » pour la stabilité. Ce lien intuitif entre les cycles positifs et les configurations stables d'un réseau a été conjecturé par Thomas, et il a fallu attendre quelques décennies pour en avoir une démonstration formelle.

Théorème 2 ([23, 19, 15])**.** *Si la dynamique d'un réseau d'automates possède plusieurs configurations stables distinctes, alors son graphe d'interaction contient un cycle positif.*

De façon analogue, dans le cadre asynchrone, il existe un lien entre les oscillations stables dans la dynamique et les cycles négatifs dans le graphe d'interaction.

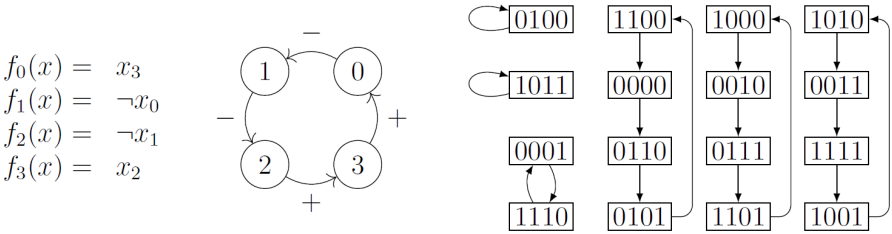

FIGURE 2. Un cycle positif à $n = 4$ automates. Fonctions locales (à gauche), graphe d'interaction (au centre), et dynamique limite (à droite, où l'on peut remarquer que toutes les $2^4 = 16$ configurations sont présentes).

Théorème 3 ([23, 18])**.** *En asynchrone, si la dynamique d'un réseau d'automates possède une oscillation stable, alors son graphe d'interaction contient un cycle négatif.*

Nous avons déjà souligné l'importance de la dynamique limite (les configurations et oscillations stables) pour les applications à la biologie. Les résultats ci-dessus donnent des conditions nécessaires à leur apparition. Pour aller plus loin dans l'étude de la dynamique d'un réseau, il convient de caractériser l'ensemble des comportements limites possibles. C'est ce qu'ont réalisé les auteurs de [7, 6], en commençant par les réseaux d'automates booléens dont le graphe d'interaction est un cycle (positif ou négatif), et pour les modes de mise à jour parallèle et asynchrone. De tels développements constituent une toute première étape dans l'étude d'une version discrète du seizième problème de Hilbert, et font déjà apparaître des formules non-triviales liées aux diviseurs de la taille du cycle (fonction indicatrice d'Euler, formule d'inversion de Möbius). Par exemple, pour un cycle positif de $n = 4$ automates, la dynamique parallèle possède 2 points fixes, 1 cycle limite de période 2, et 3 cycles limites de période 4 (voir figure 2).

L'influence des modes de mise à jour sur la dynamique limite reçoit également une attention particulière, et il s'agit d'un problème complexe. En partant du mode de mise à jour asynchrone et en autorisant une seule transition synchrone (c'est-à-dire qui met à jour l'état de plusieurs automates simultanément), la dynamique limite peut changer drastiquement (modification des trajectoires, et fusion ou destruction d'oscillations stables) [16]. Cette direction de recherche s'attaque à la sensibilité au synchronisme des réseaux d'automates ; elle est d'une grande importance pour les applications à la biologie où l'ordre de mise à jour des entités fait toujours l'objet de débat dans la communauté. On peut néanmoins affirmer que les configurations stables (points fixes dans les modes de mise à jour déterministes) bénéficient d'une

plus grande robustesse au synchronisme, car elles sont invariantes [5] pour le mode parallèle, le mode asynchrone et tous les modes de mise à jour bloc-séquentiels (où les automates sont mis à jour selon une partition ordonnée [6] de $[n]$). Cette robustesse au mode de mise à jour des configurations stables n'est toutefois pas absolue, car des modes de mise à jour plus complexes peuvent engendrer des configurations stables que l'on ne retrouve pas dans les modes classiques que sont le parallèle et l'asynchrone (par exemple avec les modes de mise à jour bloc-parallèles [17]). En résumé, il reste beaucoup de choses à comprendre sur la sensibilité des dynamiques limites à la façon dont sont mis à jour les automates au cours du temps !

L'éclairage de la théorie de la complexité

Dans cette section nous allons considérer le mode de mis à jour parallèle uniquement, et aller plus loin dans le comptage du nombre de points fixes des réseaux d'automates booléens. Des relations ont été établies entre la structure du graphe d'interaction et le nombre de points fixes de la dynamique parallèle. On parle de *bornes structurelles*.

Théorème 4 ([1, 2])**.** *Soit p le nombre de points fixes d'un réseau d'automates booléens f dont le graphe d'interaction est G. Alors $p \leqslant 2^{\tau(G)}$, où $\tau(G)$ est le nombre minimum de sommets à retirer pour rendre le graphe G acyclique. Si de plus il n'y a aucun arc signé à la fois positivement et négativement (on parle de réseau* monotone*), alors $\nu(G)+1 \leqslant p$, où $\nu(G)$ est le nombre maximum de cycles disjoints que l'on peut trouver dans G.*

En utilisant les signes des arcs du graphe d'interaction, ces travaux donnent des bornes plus précises. Par exemple, pour la borne supérieure, l'acyclicité de G peut être remplacée par l'absence de cycle positif (c'est-à-dire $p \leqslant 2^{\tau^+(G)}$, où $\tau^+(G)$ est le nombre minimum de sommets à retirer pour supprimer tous les cycles positifs de G, en laissant possiblement des cycles négatifs).

Ces bornes sont *optimales*, dans le sens où elles sont atteintes par certains réseaux d'automates booléens. On a par exemple le réseau « identité » de taille n, dont les fonctions locales sont $f_i(x) = x_i$ pour tout $i \in [n]$, dont le graphe d'interaction G est composé de n boucles, et qui possède $2^{\tau(G)}$ points fixes : il faut retirer tous les sommets de G pour supprimer tous les cycles, et toutes les configurations sont des points fixes. Pour d'autres réseaux en revanche, ces bornes structurelles sont assez

5. C'est-à-dire que pour un réseau d'automates donné, on aura le même ensemble de configurations stables pour chacun de ces modes de mise à jour.

6. Par exemple dans un réseau avec $n = 6$ automates, on peut mettre à jour les automates 0 et 1, puis les automates 2, 4 et 5, puis enfin l'automate 3, ce qui correspond à la partition ordonnée $(\{0,1\},\{2,4,5\},\{3\})$.

éloignées. Par exemple, en ajoutant un automate supplémentaire au réseau « identité » précédent, d'indice n et de fonction locale $f_n(x) = \neg x_n$, la valeur de $\tau(G)$ ne change pas mais il n'y a plus aucun point fixe car, pour toute configuration, l'état de l'automate n change. On obtient donc un réseau de $n+1$ automates avec $\tau(G) = n$, mais aucun point fixe. Ces bornes structurelles sont donc très générales, mais parfois imprécises. Savoir pour quels graphes orientés la borne supérieure est atteinte est d'ailleurs un problème fameux dans le domaine du *codage de réseau*, qui permet d'optimiser le trafic internet en mélangeant les contenus des paquets tout en étant capable de les ré-assembler lorsqu'ils arrivent à destination !

Un autre point de vue permet de comprendre pourquoi il est difficile d'affiner les bornes structurelles données par le théorème 4 : la complexité algorithmique. Si l'on ne connaît que le graphe d'interaction signé (avec les signes des arcs) à n sommets, plusieurs réseaux d'automates booléens de taille n sont possibles (et même un très grand nombre car pour chaque automate il y a une quantité doublement exponentielle en n de fonctions locales possibles). Comment calculer le nombre maximum de points fixes possibles, noté $p_{\mathsf{max}}(G)$, parmi tous ces réseaux d'automates booléens ? Existe-t-il un algorithme efficace pour calculer $p_{\mathsf{max}}(G)$ à partir de G ? La réponse est non, à moins que $\mathsf{P} = \mathsf{NP}$, c'est-à-dire à moins que de nombreux problèmes que l'on ne sait actuellement pas résoudre efficacement puissent eux aussi être tous résolus efficacement par « essentiellement » le *même* algorithme [4].

Théorème 5 ([3])**.** *Étant donné un graphe d'interaction signé G, on peut décider en temps polynomial si $p_{max}(G) \geqslant 1$, mais décider si $p_{max}(G) \geqslant k$ est NP-complet pour tout $k \geqslant 2$.*

Le théorème 5 indique que l'on peut décider avec un algorithme efficace (bien que l'algorithme en question soit très compliqué, il est « rapide ») s'il est possible d'avoir un réseau d'automates booléens dont le graphe d'interaction est G et qui a au moins un point fixe [7], mais il est par contre très difficile de décider avec un algorithme s'il est possible d'avoir un réseau d'automates booléens avec au moins deux points fixes sur le même graphe d'interaction. On peut en outre poser une question analogue sur le nombre minimum de points fixes possibles, noté $p_{\mathsf{min}}(G)$, qui est un problème encore plus complexe [8] (et pas complémentaire du problème précédent, car il existe des graphes G avec une différence arbitraire entre $p_{\mathsf{min}}(G)$ et $p_{\mathsf{max}}(G)$).

7. Pour un graphe d'interaction signé G, l'existence d'un réseau d'automates booléens possédant au moins un point fixe est démontré équivalent à la présence d'un cycle positif dans G, ce qui peut être testé en temps polynomial par un algorithme qui a mis environ 20 ans à être découvert et est hautement non-trivial...

8. NEXPTIME est une classe de complexité caractérisant des problèmes très très difficiles à résoudre algorithmiquement (c'est-à-dire qui sont résolus par des algorithmiques très peu efficaces : en temps exponentiel).

Théorème 6 ([3]). *Étant donné un graphe d'interaction signé G, décider si* $p_{min}(G) \leqslant k$ *est* NEXPTIME*-complet pour tout* $k \geqslant 1$.

Puisqu'il est même difficile de calculer le nombre maximum et minimum de points fixes parmi les réseaux d'automates booléens sur *un* graphe d'interaction donné, il est inévitable que des bornes précises et générales fassent appel à des propriétés structurelles très complexes du graphe d'interaction. Les constructions utilisées pour démontrer les théorèmes 5 et 6 possèdent un entrelacement délicat entre les cycles positifs et négatifs des graphes d'interactions, qui esquisse à peine la grande subtilité des propriétés en jeu...

Pour conclure

Au cours de ce bref survol de thématiques de recherches fondamentales sur le modèle des réseaux d'automates, nous avons rencontré des problèmes abstraits aux applications vastes, mais qui s'avèrent très compliqués à résoudre. Ce domaine est tiré par des questions de l'ordre de *graals*, comme la caractérisation des points fixes et des cycles limites faisant écho à l'un des 23 problèmes de Hilbert. Il y a pour l'heure peu de pistes de résolution complète de ces questionnements, mais nous avons vu que la complexité algorithmique apporte un regard nouveau qui *explique* en partie pourquoi il ne semble pas y avoir de réponses simples à ces questions complexes. Le modèle mathématique des réseaux d'automates permet toutefois de progresser par étapes, suivant une approche constructive consistant à commencer par des études restreintes, qui sont petit-à-petit composées pour aller en direction de cas plus généraux. Ainsi, même si les grandes interrogations qui portent le sujet sur le temps long sont toujours ouvertes, des progrès significatifs n'ont eu de cesse d'être actés. Ces résultats nourrissent l'enthousiasme d'une large communauté aux multiples facettes !

Références

[1] Julio Aracena. Maximum number of fixed points in regulatory Boolean networks. *Bulletin of Mathematical Biology*, 70 :1398–1409, 2008.

[2] Julio Aracena, Adrien Richard, and Lilian Salinas. Number of fixed points and disjoint cycles in monotone Boolean networks. *SIAM Journal on Discrete Mathematics*, 31 :1702–1725, 2017.

[3] Florian Bridoux, Amélia Durbec, Kévin Perrot, and Adrien Richard. Complexity of fixed point counting problems in Boolean Networks. *Journal of Computer and System Sciences*, 126 :138–164, 2022.

[4] Jean-Paul Delahaye. P = NP, un problème à un million de dollars ? *Interstices*, 2007.

[5] Jean-Paul Delahaye and François Rechenmann. La simulation par ordinateur change-t-elle les sciences ? *Interstices*, 2006.

[6] Jacques Demongeot, Tarek Melliti, Mathilde Noual, Damien Regnault, and Sylvain Sené. *Automata and complexity*, volume 42 of *Springer Series on Emergence, Complexity, Computation*, chapter On Boolean automata isolated cycles and tangential double-cycles dynamics, pages 145–178. Springer, 2022.

[7] Jacques Demongeot, Mathilde Noual, and Sylvain Sené. Combinatorics of Boolean automata circuits dynamics. *Discrete Applied Mathematics*, 160 :398–415, 2012.

[8] Ben Hesper and Paulien Hogeweg. Bioinformatica: een werkconcept. *Het Kameleon*, 1 :28–29, 1970.

[9] François Jacob and Jacques Monod. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, 3 :318–356, 1961.

[10] Stuart A. Kauffman. *Current topics in developmental biology*, volume 6, chapter Gene regulation networks: A theory for their global structures and behaviors, pages 145–181. Elsevier, 1971.

[11] Stephen C. Kleene. *Automata Studies*, volume 34, chapter Representation of events in nerve nets and finite automata, pages 3–42. Princeton University Press, 1956.

[12] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *Journal of Mathematical Biophysics*, 5 :115–133, 1943.

[13] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969.

[14] John von Neumann. *Theory of self-reproducing automata*. University of Illinois Press, 1966.

[15] Mathilde Noual. *Updating Automata Networks*. PhD thesis, École normale supérieure de Lyon, 2012.

[16] Mathilde Noual and Sylvain Sené. Synchronism versus asynchronism in monotonic Boolean automata networks. *Natural Computing*, 17 :393–402, 2018.

[17] Loïc Paulevé and Sylvain Sené. *Systems biology modelling and analysis : formal bioinformatics methods and tools*, chapter Boolean networks and their dynamics : the impact of updates. Wiley. À paraître.

[18] Adrien Richard. Negative circuits and sustained oscillations in asynchronous automata networks. *Advances in Applied Mathematics*, 44 :378–392, 2010.

[19] Adrien Richard and Jean-Paul Comet. Necessary conditions for multistationarity in discrete dynamical systems. *Discrete Applied Mathematics*, 155 :2403–2413, 2007.

[20] François Robert. Itérations sur des ensembles finis et automates cellulaires contractants. *Linear Algebra and its Applications*, 29 :393–412, 1980.

[21] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65 :386–408, 1958.

[22] René Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42 :563–585, 1973.

[23] René Thomas. On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations. In *Numerical methods in the study of critical phenomena*, volume 9, pages 180–193. 1981.

[24] Alan Turing. On Computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society. Second Series*, 45 :230–265, 1937.