Sujet de stage – M2
Synchronisation des réseaux booléens asynchrones

Titre : Synchronisation des réseaux booléens asynchrones.

Encadrants : Florian Bridoux et Adrien Richard — `richard@unice.fr` — page-perso.

Lieu : Laboratoire I3S, UMR CNRS 71-271, Université Côte d'Azur, Sophia Antipolis.

Sujet : Les *réseaux d'automates* modélisent des ensembles *finis* d'entités simples évoluant dans un *temps discret* et sur des *domaines finis*, selon des règles d'évolution *locales*. Ils trouvent de nombreuses applications – en biologie, sociologie ou informatique – car, malgré leur simplicité, ils engendrent une grande diversité de comportements dynamiques, faisant d'eux un cadre unificateur pour l'étude des systèmes complexes. Leur théorie relie des domaines variés – théorie des graphes, de la complexité, combinatoire, entre autres – et pose des questions fondamentales sur les relations entre structure, règles locales et comportements émergents.

On considère dans ce stage les *réseaux booléens*, c'est-à-dire les réseaux d'automates à n entités binaires, notées de 1 à n. Un tel réseau est décrit à partir d'une fonction f de l'ensemble des vecteurs binaires à n composantes dans lui-même :

$$f : \{0,1\}^n \to \{0,1\}^n, \qquad x = (x_1, \dots, x_n) \mapsto f(x) = (f_1(x), \dots, f_n(x)).$$

La fonction locale de l'entité i est la i-ème composante de f, notée f_i, qui est une fonction booléenne à n composantes, de $\{0,1\}^n$ dans $\{0,1\}$.

On étudie souvent la dynamique *synchrone* de f, décrite par les itérations de f. Autrement dit, à chaque étape de temps, toutes les fonctions locales sont appliquées simultanément. Mais dans bien des applications – et en particulier dans le contexte des réseaux de gènes – il est plus réaliste d'appliquer, à chaque étape de temps, une et une seule fonction locale : on parle de dynamique *asynchrone*. Plus précisément, appliquer la fonction locale f_i en x, c'est passer au vecteur $(x_1, \dots, f_i(x), \dots, x_n)$, que l'on note $f^i(x)$. Ainsi, f^i est un réseau booléen à n composantes.

La collection des f^i forme un automate fini déterministe (AFD), appelé *dynamique asynchrone* de f :
- l'alphabet est $\{1, \dots, n\}$,
- l'ensemble des états est $\{0,1\}^n$,
- la transition étiquetée par i partant de x mène à $f^i(x)$.

Étant donné un mot sur l'alphabet $\{1, \dots, n\}$, on note $f^w(x)$ l'état que l'on atteint depuis x en lisant le mot w dans l'automate fini déterministe : si $w = i_1, i_2, \dots, i_k$ alors $f^w(x) = (f^{i_k} \circ \dots \circ f^{i_2} \circ f^{i_1})(x)$. Ainsi, f^w est encore un réseau booléen à n composantes. On dit que w *synchronise* f si f^w est une fonction constante : quel que soit l'état initial, la lecture de w mène toujours au même état.

La synchronisation est un problème profond et très étudié. La conjecture la plus célèbre du domaine est la conjecture de Černý (1964) : si un AFD à q états a un mot synchronisant, alors il en a un de longueur au plus $(q-1)^2$. Le stage consiste à étudier cette conjecture dans le cas particulier des réseaux booléens : si f a un mot synchronisant, alors il en a un de longueur au plus $(2^n - 1)^2$.

Ce cas particulier est intéressant à plusieurs points de vue. Les réseaux booléens ont de nombreuses applications, et leur synchronisation pourrait être une propriété cruciale dans certains cas. Ils sont par ailleurs très structurés : la dynamique asynchrone est essentiellement une orientation de l'hypercube, structure qui se prête particulièrement bien aux raisonnements par induction. De plus, la conjecture a été démontrée pour certaines familles d'AFDs [1], mais les réseaux booléens diffèrent complètement de ces familles ; leur étude pourrait donc ajouter une (très) nouvelle pierre à l'édifice. Enfin, quelques résultats partiels sont encourageants [2] et pourront servir de point de départ.

[1] Mikhail V Volkov. Synchronizing automata and the Černý conjecture. In International Conference on Language and Automata Theory and Applications, pages 11–27. Springer, 2008. lien pdf

[2] Synchronizing Boolean networks asynchronously, J. Aracena, A. Richard, L. Salinas. Journal of Computer and System Sciences, 136 :249-279, 2023. lien arXiv