

RAPPORT DE STAGE DE L3

ANALYSE DE LA DYNAMIQUE DES
MODÈLES BIOLOGIQUES PAR
PROGRAMMATION LOGIQUE

Léo-Paul DELSAUX[†]

Encadrant du stage
Maxime FOLSCHETTE[‡]

Juin-Août 2022

[†]ENS de Lyon

[‡]Équipe Bio-Computing, laboratoire CRIStAL, CNRS de Lille

Table des matières

1	Introduction	2
2	Answer Set Programming	2
3	Modèle	3
3.1	Avantages du Modèle	3
3.2	Modèle en ASP	3
4	Définitions	3
5	Lemmes	3
6	Algorithme	3
7	Conclusion	3

1 Introduction

La biologie s'intéresse à l'étude des systèmes vivants ou composant le vivant (gènes, cellules, écosystèmes...). La bioinformatique est l'utilisation de méthodes informatiques pour aider la biologie. Un des domaines de la bioinformatique s'appelle la *biologie des systèmes* et consiste à proposer des modèles et des méthodes informatiques et mathématiques pour représenter et étudier de tels systèmes.

Ainsi, des études biologiques permettent de déterminer comment certains gènes interagissent entre eux, ce qui peut être représenté mathématiquement sous la forme d'un graphe. De plus, ces interactions peuvent être vues comme des transitions entre les sommets de notre graphe, ce qui se rapproche alors plus d'un automate. La représentation du modèle étudié lors de ce stage est explicité dans le chapitre Modèle.

L'étude de la dynamique de systèmes biologiques lève plusieurs problèmes tels que l'identification d'attracteurs, les bifurcations ou encore la connexité entre deux états globaux. Dans ce rapport je vais discuter de la recherche d'attracteurs dans des réseaux d'automates asynchrones : il s'agit d'un ensemble d'états duquel on ne peut pas s'échapper et minimal au sens de l'inclusion.

Ce qui sera étudié dans ce rapport ne le sera que partiellement : on n'entrera pas en détail dans le code présenté, mais on se penchera simplement sur les grandes lignes et sur les points techniques qu'il arbore.

2 Answer Set Programming

L'Answer Set Programming (ASP) est un paradigme de programmation comparable à Prolog. Ces dernières décennies, ASP s'est trouvé être paradigme de programmation logique puissant pour traiter des modèles biologiques, permettant de parcourir un grand nombre de configurations rapidement. ASP peut énumérer facilement un grand nombre d'ensembles solutions au problème encodé : c'est un paradigme très efficace pour la combinatoire. Nous allons ici présenter brièvement son fonctionnement, et plus précisément, les outils qui ont été utiles pour mon stage.

Un *programme d'ensemble solutions* (= *answer set program*) est un nombre fini de règles de la forme :

$$a_0 \text{ :- } a_1, \dots, a_m, \text{not } a_{m+1}, \dots, \text{not } a_n.$$

avec $n \geq m \geq 0$, a_0 est un **atome** ou \perp (Bottom/le Faux) et représente ici la **tête** de la règle. Les a_1, \dots, a_n sont des atomes et représente quant à eux le **corps** de la règle, et le symbole "*not*" représente la négation par l'échec. Cette règle se lit intuitivement : si les atomes a_1, \dots, a_m sont tous vrais et qu'aucun des atomes a_{m+1}, \dots, a_n n'est vrai, alors a_0 est vrai.

Si $n = m = 0$, cela signifie que a_0 est vrai. Dans ce cas, on parle d'un fait, et

on ne doit pas renseigner ":-". D'une autre part, si $a_0 = \perp$, on parle de contrainte : comme \perp ne peut jamais être vrai, si le corps de la règle est vrai, cela invalide la solution actuelle. On ne renseigne pas \perp pour la tête, on laisse une tête vide au niveau du code.

Cette notion de "solution actuelle" se formalise de la façon suivante : on parle d'une **interprétation** I en tant qu'ensemble fini d'atomes propositionnels. Une règle r définie comme ci-dessus est *vraie dans* I si et seulement si :

$$\{a_1, \dots, a_m\} \subseteq I \wedge \{a_{m+1}, \dots, a_n\} \cap I = \emptyset \Rightarrow a_0 \in I$$

Si toutes les règles d'un programme P sont vraies dans une même interprétation I , alors on dit que I est un **modèle** de P

3 Modèle

On définit ce qu'est un réseau d'automates asynchrones proprement.

3.1 Avantages du Modèle

Très simple à utiliser en pratique.

3.2 Modèle en ASP

Un bout de code

4 Définitions

état local, sémantique, état global, état stable, domaines de pièges, attracteurs.

5 Lemmes

6 Algorithme

7 Conclusion