

# Table des matières

Acknowledgement	2
Abstract	3
1 Process Hitting	5
2 Conclusion et perspectives	7

# Acknowledgement

The internship opportunity I had with National Institute of Informatics (NII) was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me through this internship period.

I express my deepest thanks to Dr. Pr. Katsumi Inoue, my team leader and my supervisor for taking part in useful decision & giving necessary advices and guidance and arranged all facilities to make life easier. I choose this moment to acknowledge his contribution gratefully.

Bearing in mind previous I am using this opportunity to express my deepest gratitude and special thanks to my Ph.D. supervisor Ass. Pr. Morgan Magnin who in spite of being extraordinarily busy with his duties, took time out to hear, guide and keep me on the correct path and allowing me to carry out my researches at their esteemed organization and extending during the training.

It is my radiant sentiment to place on record my best regards, deepest sense of gratitude to my supervisor Dr. P. Olivier Roux at France for his careful and precious guidance which were extremely valuable for my study both theoretically and practically. Also I want to thank him and thank Ass. P Morgan Magnin for their effort to allow me to do this interesting internship.

I also would like to thank all the people that worked in the office of NII in Tokyo. With their patience and openness they created an enjoyable working environment. Furthermore I want to thank all the scientist and students, with whom I did the fieldwork. We experienced great things together.

I perceive as this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future.

# Abstract

Almost the models of biological networks are not robust and need some times to be revised and adapted to the new observations. A system maintains its functions against internal and external perturbations, leading to topological changes in the network with varying delays. To understand the resilient behaviour of biological networks, we propose novel methods. First we propose an approach to model a time-dependent asynchronous and non-deterministic networks through Process Hitting (PH) framework which is a new framework particularly suitable to model biological regulatory networks. Second we have developed a novel network completion algorithm for time-varying networks to analyse its behavior based on the framework of network completion. This completion aims to make the minimum amount of modifications to a given network so that the resulting network is most consistent with the observed data. We demonstrate the effectiveness of our proposed methods through computational experiments using synthetic gene expression data of the circadian clock network modeled through PH. The results indicate that our methods exhibit good performance in terms of completing and inferring gene association networks with time-varying structures.

# Introduction

# 1 Process Hitting

Definition 1 introduces the Process Hitting(PH) [?] which allows to model a finite number of local levels, called *processes*, grouped into a finite set of components, called *sorts*. A process is noted  $a_i$ , where  $a$  is the sort's name, and  $i$  is the process identifier within sort  $a$ . At any time, exactly one process of each sort is *active*, and the set of active processes is called a *state*.

The concurrent interactions between processes are defined by a set of *actions*. Each action is responsible for the replacement of one process by another of the same sort conditioned by the presence of at most one other process in the current state. An action is denoted by  $a_i \rightarrow b_j \uparrow b_k$ , which is read as “ $a_i$  hits  $b_j$  to make it bounce to  $b_k$ ”, where  $a_i, b_j, b_k$  are processes of sorts  $a$  and  $b$ , called respectively *hitter*, *target* and *bounce* of the action. We also call a *self-hit* any action whose hitter and target sorts are the same, that is, of the form :  $a_i \rightarrow a_i \uparrow a_k$ .

The PH is therefore a restriction of synchronous automata, where each transition changes the local state of exactly one automaton, and is triggered by the local states of at most two distinct automata. This restriction in the form of the actions was chosen to permit the development of efficient static analysis methods based on abstract interpretation [?].

**Definition 1** (Process Hitting). A *Process Hitting* is a triple  $(\Sigma, \mathcal{L}, \mathcal{H})$  where :

- $\Sigma = \{a, b, \dots\}$  is the finite set of *sorts* ;
- $\mathcal{L} = \prod_{a \in \Sigma} \mathcal{L}_a$  is the set of *states* where  $\mathcal{L}_a = \{a_0, \dots, a_{l_a}\}$  is the finite set of *processes* of sort  $a \in \Sigma$  and  $l_a$  is a positive integer, with  $a \neq b \Rightarrow \mathcal{L}_a \cap \mathcal{L}_b = \emptyset$  ;
- $\mathcal{H} = \{a_i \rightarrow b_j \uparrow b_k \in \mathcal{L}_a \times \mathcal{L}_b^2 \mid (a, b) \in \Sigma^2 \wedge b_j \neq b_k \wedge a = b \Rightarrow a_i = b_j\}$  is the finite set of *actions*.

*Example.* The figure 1 represents a  $\mathcal{PH}(\Sigma, \mathcal{L}, \mathcal{H})$  with three sorts ( $\Sigma = \{a, b, c\}$ ) and :  $\mathcal{L}_a = \{a_0, a_1\}$ ,  $\mathcal{L}_b = \{b_0, b_1\}$ ,  $\mathcal{L}_c = \{z_0, z_1, z_2\}$ .

A state of the networks is a set of active processes containing a single process of each sort. The active process of a given sort  $a \in \Sigma$  in a state  $s \in \mathcal{L}$  is noted  $s[a]$ . For any given process  $a_i$  we also note :  $a_i \in s$  if and only if  $s[a] = a_i$ .

**Definition 2** (Playable action). Let  $\mathcal{PH} = (\Sigma, \mathcal{L}, \mathcal{H})$  be a Process Hitting and  $s \in \mathcal{L}$  a state of  $\mathcal{PH}$ . We say that the action  $h = a_i \rightarrow b_j \uparrow b_k \in \mathcal{H}$  is *playable in state*  $s$  if and only if  $a_i \in s$  and  $b_j \in s$  (i.e.  $s[a] = a_i$  and  $s[b] = b_j$ ). The resulting state after playing  $h$  in  $s$  is called a *successor* of  $s$  and is denoted by  $(s \cdot h)$ , where  $(s \cdot h)[b] = b_k$  and  $\forall c \in \Sigma, c \neq b \Rightarrow (s \cdot h)[c] = s[c]$ .

The PH was chosen for several reasons. First, it is a general framework that, although it was mainly used for biological networks, allows to represent any kind of dynamical model, and converters to several other representations are available (see section). Although an efficient dynamical analysis already exists for this framework, based on an approximation of the dynamics, it is interesting to identify its limits

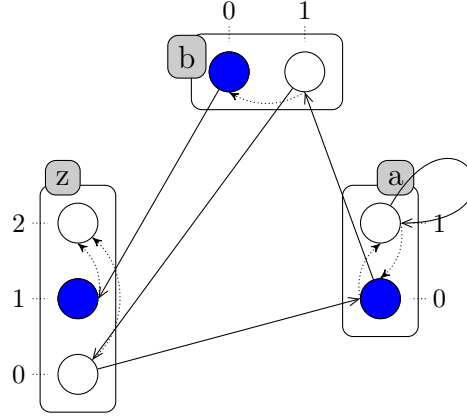


FIGURE 1 – A PH model example with three sorts :  $a$ ,  $b$  and  $z$  ( $a$  is either at level 0 or 1,  $b$  at either level 0 or 1 and  $z$  at either level 0, 1 or 2). Boxes represent the *sorts* (network components), circles represent the *processes* (component levels), and the 5 *actions* that model the dynamic behavior are depicted by pairs of arrows in solid and dotted lines. The grayed processes stand for the possible initial state :  $\langle a_1, b_0, z_1 \rangle$ .

and compare them to the approached we present later in this paper. Finally, the particular form of the actions in a PH model allow to easily represent them in ASP, with one fact per action, as described in the next section. Other representations may have required supplementary complexity ; for instance, a labeling would be required if actions could be triggered by a variable number of processes.

**Definition 3** (Action). An action is noted  $a_i \rightarrow b_j \uparrow b_k$  where  $a_i$  is a process of sort  $a$  and  $b_j, b_k$  two processes of sort  $b$ . When  $a_i = b_j$ , such an action is referred as a self-action and  $a_i$  is called a self-hitting process.

**Definition 4** (Conjoint action).

**Definition 5** (timed action).

The PH was chosen for several reasons. First, it is a general framework that, although it was mainly used for biological networks, allows to represent any kind of dynamical model, and converters to several other representations are available (see section). Although an efficient dynamical analysis already exists for this framework, based on an approximation of the dynamics, it is interesting to identify its limits and compare them to the approached we present later in this paper. Finally, the particular form of the actions in a PH model allow to easily represent them in ASP, with one fact per action, as described in the next section. Other representations may have required supplementary complexity ; for instance, a labeling would be required if actions could be triggered by a variable number of processes.

The rest of the paper focuses on the resolution of the previous issues with the use of ASP. The enumeration of all fixed points of a PH model will be tackled in section and the verification of a reachability property will be the subject of section

## 2 Conclusion et perspectives

Nous avons montré dans cette thèse de Master une nouvelle analyse dynamique développée. Cette analyse est applicable à une classe de modèles dits PH et vise à déterminer des propriétés des réseaux modélisés.

La première propriété est la recherche des états stables du réseau qu'on appelle les points fixes. Ces points représentent les états du réseau pendant lesquels, le modèle ne peut plus évoluer. Il est intéressant à les connaître car ils bloquent l'évolution des systèmes biologiques. Nous avons développé deux méthodes, sachant que la deuxième est plus optimale, qui retournent l'ensemble de tous les points fixes du réseau. Ce point fixe n'est qu'un état du réseau traduit en ASP par un niveau pour chaque composant, autrement un processus pour chaque sorte.

La deuxième propriété est une propriété qui se base sur la dynamique du réseau : l'atteignabilité. Un réseau biologique évolue dans plusieurs sens qui peuvent aboutir ou pas à des états-objectifs. Notre nouvelle approche retourne les chemins exactes aboutissant à atteindre un niveau cible d'un composant du système. Ce chemin se traduit par un ensemble de changements successives des niveaux. En PH cela se traduit par de changements successifs de processus et c'est ce que nos méthodes retournent comme résultat. Il s'avère que la méthode itérative en ASP est plus optimale que celle en ASP normal. En effet il n'est pas nécessaire de prévoir le nombre de changements pour la méthode itérative et elle retourne le résultat plus rapidement (quelques seconde pour des réseau moyennement grand).

Une comparaison a été faite par rapport à l'existant, PINT et la méthode de Rocca et al. Les résultats montrent que, par rapport à PINT, la méthode de recherche des points fixes est efficace, mais que pour l'accessibilité, elle l'est moins que prévu. Cependant aussi notre méthode retourne de plus le chemin d'atteignabilité. Elle offre la possibilité de poser des questions plus générales par rapport à PINT portant sur plusieurs sortes de plus.

Sachant que la présentation d'un réseau biologique en PH est simplifiée le traitement et la traduction des modèles, il s'avère que notre méthode qui se base sur ce formalisme est plus efficace que d'autres méthodes développées en ASP aussi mais pour des réseaux de graphes de transitions. Le cas de la méthode de Rocca qui est gourmande en temps par rapport à la notre, résultat retourné en des minutes contre un résultat affiché en quelques secondes.

Nous pensons que cette approche peut également être utilisée et adaptée avec d'autres modèles tels que le modèle de Thomas, les réseaux de Petri et les modèles synchrones. Cela nécessite une traduction propre au modèle étudié ainsi qu'un traitement approprié.

Parmi nos perspectives qui font partie de mon sujet de thèse, c'est d'essayer d'améliorer cette méthode en éliminant les cycles de la méthode itérative. Cela évite de tourner indéfiniment dans des boucles sans avoir un résultat affiché.

Ensuite, nous souhaitons étendre le programme pour chercher les attracteurs. Un attracteur est un ensemble d'états à partir desquels il n'est plus possible de sortir,

et donc tel que le réseau tourne indéfiniment dans ces états. Le point fixe est un cas spécial des attracteur, en effet c'est un attracteur de dimension une. Par contre, la caractérisation des attracteurs de la dynamique, de dimension  $n$ , requiert une analyse des dynamiques possibles bien plus poussée que pour les points fixes.

Nous visons aussi à implémenter une recherche dynamique dans le sens inverse de l'atteignabilité et poser la question : "*Quels sont les états initiaux qui nous permettent d'atteindre nos objectifs ?*". La réponse à cette question est l'ensemble des états à partir desquels il existe des chemins qui activent le ou les objectif(s). C'est vrai que la méthode de Rocca et al. résolve cette problématique mais nous estimons à avoir une approche plus efficace en terme de temps et qui utilise le réseau en process hitting et non pas les graphes de transitions.

Tous ces problématiques constituent une perspective intéressante dans le cadre du développement de techniques d'analyse statique et dynamiques des propriétés du Process Hitting.