Answer Set Programming Method for Network Completion for Time-Varying Genetic Networks modeled in Process Hitting

Emna Ben Abdallah[1], Tony Ribeiro[2], Morgan Magnin[1,2], and Katsumi Inoue[1]

[1] LUNAM Universit, École Centrale de Nantes, IRCCyN UMR CNRS 6597
(Institut de Recherche en Communications et Cybernétique de Nantes),
1 rue de la No, 44321 Nantes, France.
[2] National Institute of Informatics,
2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan.

Abstract. IAlmost 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 func- tions 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 algo- rithm 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 circadien 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.

Keywords: Answer Set Programming, Process Hitting, time-varying genetic networks, network completion

1 Introduction

2 Process Hitting

Definition 1 introduces the Process Hitting(PH) [1] 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 $a_i \rightarrow b_j \Rsh b_k$, 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: $a_i \rightarrow a_i \Rsh a_k$.

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 [2].

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* $a \neq b \Rightarrow \mathcal{L}_a \cap \mathcal{L}_b = \emptyset$*;*
- $\mathcal{H} = \{a_i \rightarrow b_j \Rsh b_k \in \mathcal{L}_a \times \mathcal{L}_b^2 \mid (a, b) \in \Sigma^2 \wedge b_j \neq b_k \wedge a = b \Rightarrow a_i = b_j\}$ *is the finite set of* actions*.*

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

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 $s[a] = a_i$. The dynamic of the PH networks is satisfied thanks to the actions. Indeed, the transition from one state s_1 to its successor s_2 is done when there is a playable action (definition 2) at s_1. After each transition only one sort, or one component, changes its level from one process to another.

Definition 2 (Playable action). *Let* $\mathcal{PH} = (\Sigma, \mathcal{L}, \mathcal{H})$ *be a Process Hitting and* $s \in \mathcal{L}$ *a state of PH. We say that the action* $h = a_i \rightarrow b_j \Rsh b_k \in \mathcal{H}$ *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* $\forall c \in \Sigma, c \neq b \Rightarrow (s \cdot h)[c] = s[c]$*.*

We note that during these last years the Process Hitting framework was improved and we added new type of sorts like cooperative sorts and new actions like plural actions, actions with priority and actions with delay.

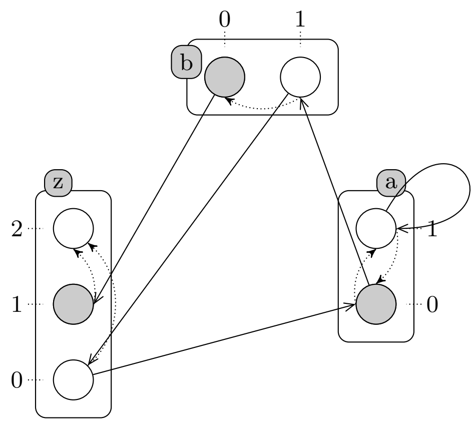

Fig. 1. A PH model example with three sorts: a, b and z (a is either at level 0 or 1, b at either level 0 or 1 and z at either level 0, 1 or 2). Boxes represent the *sorts* (network components), circles represent the *processes* (component levels), and the 5 *actions* that model the dynamic behavior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

In some cases it is necessary to represent a reaction of a set of components on one component. For example in the bio-chemical reactions :$X \xrightarrow{Y} Z$ or $X + Y \rightarrow Y + Z$, where X is a set of reactives, Y a set of catalysts and Z a set of products.The plural action permits to represent this kind of reactions in PH. The plural is made up of two sets of processes of different sorts, which represent all the hitters and the bonds.

Definition 3 (Plural action). *It is a reaction of this form:*
$A \rightarrowtail B \mid A, B \in Proc \setminus \emptyset \wedge \forall q \in B, \exists p \in A, (p \neq q \wedge \Sigma(p) = \Sigma(q))$

Example 2. We give a simple example to represent a plurial action by a cooperation between two biological components (x and y) in order to activate another component (z) and change its level from 0 to 1: $\{x_1, y_1, z_0\} \rightarrow \{x_1, y_1, z_1\}$.

In some dynamics it is crucial to have information about the delays between two events (two states in PH). The normal actions cannot show this information we just know that the state s_2 will be after s_1 in the next step but it is not possible to know how much time this transition takes time. We propose to add the delay in the action attributes which is responsable of the transition between the two states. That means that this action needs to be played during a specific time so that the system doesn't change the state (defintion 4).

Definition 4 (timed action).

Definition 5 (temporised sort).

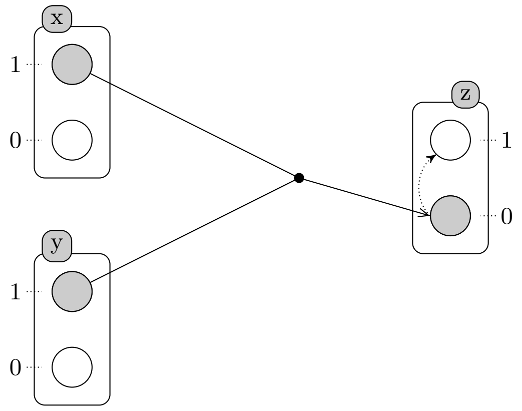

Fig. 2. Representation of a plural action in Process Hitting network: $\{x_1, y_1, z_0\} \rightarrow \{x_1, y_1, z_1\}$.

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 sectionAlthough 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 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 sectionAlthough 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 third 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 report focuses on the representation of the previous definitions through ASP than we give the example of Circadian Clock network. Later we propose an approch to resolve the completion problem of PH networks with the use of ASP.

3 PH through ASP

3.1 Translation of PH networks to ASP

3.2 Circadian clock in ASP

4 Completion of PH networks

4.1 Asynchronous and non-deterministic networks

4.2 Algorithm

- INPUT: a Process Hitting and a chronogram of the genes evolution.
- Step 1: Sample the chronogram for each gene at each time step.
- Step 2: For each time step where a gene G changes its value from val to val':
  - 2.1: Compute D, the delay since the last time a gene has changed.
  - 2.2: Generate all actions with delay D which involve all subsets of the genes $X_1, \ldots, X_n$ having an influence on G:

$$action(X_1, Val_1, \ldots, X_n, Val_n, G, val, val', D)$$

- Step 3: For each action with the same hitters, $X_1, Val_1, \ldots, X_n, Val_n$ and the same target, G, val, val', merge into one action where the delay is the average.
- OUTPUT: a completed Process Hitting that realize the chronogram.

Theorem 1 (Complexity). *Let PH be a Process Hitting, S be the number of sorts of PH and P be the maximal number of processes of a sort of PH. Let C be a chronogram of the genes of PH over T units of time. The complexity of completing PH by learning actions from the observations of C with our algorithm belongs to $O(T * P^{S+1})$.*

Proof. Let i be the maximal indegree of an action in PH, $0 \leq i \leq P$. Let p be a process of PH and n be the number of sorts that can influence p. There is i^S possible combinations of those process that can hit p, each of those can form an action. Since there is P process, there are $P * i^S$ possibles actions, thus the memory of our algorithm is bound by $O(P * i^S)$, which belongs to $O(P^{S+1})$ since $0 \leq i \leq P$.

Sampling the chronogram (step 1) is linear in the number of time step and then bound by $O(T)$. At each time step, atmost one gene can change its value in the chronogram C. Thus atmost i^S actions can be produced at each time step, the complexity is then bound by $O(T * i^S)$. Since $0 \leq i \leq P$, the complexity of generating actions from the observations of C (Step 2) belongs by $O(T * P^S)$. Merging the actions (step 3) is linear in the number of actions and then bound by $O(P^{S+1})$. So, finally, the complexity of completing PH by learning actions from the observations of C with our algorithm is $O(T + T * i^S + P^{S+1})$, which belongs to $O(T * P^{S+1})$

□

5 Conclusion and perspectives

References

1. Loïc Paulevé, Morgan Magnin, and Olivier Roux. Refining dynamics of gene regulatory networks in a stochastic π-calculus framework. In *Transactions on Computational Systems Biology XIII*, pages 171–191. Springer, 2011.
2. Loïc Paulevé, Morgan Magnin, and Olivier Roux. Static analysis of biological regulatory networks dynamics using abstract interpretation. *Mathematical Structures in Computer Science*, 22(04):651–685, 2012.