Concretizing the Process Hitting into Biological Regulatory Networks

Maxime Folschette[1,2], Loïc Paulevé[3], Katsumi Inoue[2], Morgan Magnin[1], Olivier Roux[1]

[1] LUNAM Université, École Centrale de Nantes, IRCCyN UMR CNRS 6597
(Institut de Recherche en Communications et Cybernétique de Nantes)
1 rue de la Noë - B.P. 92101 - 44321 Nantes Cedex 3, France.
Maxime.Folschette@irccyn.ec-nantes.fr
[2] National Institute of Informatics,
2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan.
[3] LIX, École Polytechnique, 91128 Palaiseau Cedex, France.

Abstract. The Process Hitting (PH) is a recently introduced framework to model concurrent processes. Its major originality lies in a specific restriction on the causality of actions, which makes the formal analysis of very large systems tractable. PH is suitable to model Biological Regulatory Networks (BRNs) with complete or partial knowledge of cooperations between regulators by defining the most permissive dynamics with respect to these constraints.

On the other hand, the qualitative modeling of BRNs has been widely addressed using René Thomas' formalism, leading to numerous theoretical work and practical tools to understand emerging behaviors.

Given a PH model of a BRN, we first tackle the inference of the underlying Interaction Graph between components. Then the inference of corresponding Thomas' models is provided using Answer Set Programming, which allows notably an efficient enumeration of (possibly numerous) compatible parametrizations.

In addition to giving a formal link between different approaches for qualitative BRNs modeling, this work emphasizes the ability of PH to deal with large BRNs with incomplete knowledge on cooperations, where Thomas' approach fails because of the combinatorics of parameters.

1 Introduction

As regulatory phenomena play a crucial role in biological systems, they need to be studied accurately. Biological Regulatory Networks (BRNs) consist in sets of either positive or negative mutual effects between the components. With the purpose of analyzing these systems, they are often modeled as graphs which make it possible to determine the possible evolutions of all the interacting components of the system. Indeed, besides continuous models of physicists, often designed through systems of ordinary differential equations, a discrete modeling approach was initiated by René Thomas in 1973 [1].

In this approach, the different levels of a component, such as concentration or expression levels, are abstractly represented by (positive) integer values and transitions between these levels may be considered as instantaneous. Hence, qualitative state graphs may be derived from which we are able to formally find out all the possible behaviors expressed as sequences of transitions between these states. Nevertheless, these dynamics can be precisely established only with regard to some discrete parameters, hereafter called "Thomas' parameters", which stand for kinds of "focal points", i.e. the evolutionary tendency from each state and depending on the set of the other currently interacting components.

Thomas' modeling has motivated numerous works around the link between the Interaction Graph (IG) (summarizing the global influences between components) and the possible dynamics (e.g., [2,3]), model reduction (e.g., [4]), formal checking of dynamics (e.g., [5,6]), and the incorporation of time (e.g., [7,8]) and probability (e.g., [9]) dimensions, to name but a few. While the formal checking of dynamical properties is often limited to small networks because of the state graph explosion, the main drawback of this framework is the difficulty to specify Thomas' parameters, especially for large networks.

In order to address the formal checking of dynamical properties within very large BRNs, we recently introduced in [10] a new formalism, named the *"Process Hitting"* (PH), to model concurrent systems having components with a few qualitative levels. A PH describes, in an atomic manner, the possible evolutions of a process (representing one component at one level) triggered by the hit of at most one other process in the system. This framework can be seen as a special class of formalisms like Petri Nets or Communicating Finite State Machines, where the causality between actions is restricted. Thanks to the particular structure of interactions within a PH, very efficient static analysis methods have been developed to over- and under-approximate reachability properties making tractable the formal analysis of BRNs with hundreds of components [11].

PH is suitable to model BRNs with different levels of abstraction in the specification of cooperations (associated influences) between components. This allows to model BRNs with a partial knowledge on precise evolution functions for components by capturing the largest (the most general) dynamics.

The objectives of the work presented in this paper are the following. Firstly, we show that starting from one PH model, it is possible to find back the underlying IG. We perform an exhaustive search for the possible interactions on one component from all the others, consistently with the knowledge of the dynamics that these interactions lead to and that are expressed in PH. The second phase of our work concerns the Thomas' parameters inference. It consists in determining the nesting set (possibly too large) of the parameters which necessarily lead to the satisfaction of the known cooperating constraints. The resulting BRN dynamics is ensured to respect the PH dynamics, i.e. no spurious transitions are made possible by the inference. Answer Set Programming (ASP) [12] turns out to be effective for these enumerative searches.

The outcome of this work is twofold. The first benefit is that such an approach makes it possible to refine the construction of BRNs with a partial and

progressively brought knowledge in PH, while being able to export such models in the Thomas' framework. This work thus strengthens the formal link between both modelings. The second feature of our method is that it can be applied on very large BRNs.

Finally, it must be noticed that we are not interested in this paper in the derivation of one PH from a BRN (which was previously described in [10]) but, on the contrary, to finding out a set of BRNs from one PH.

Our work is related to the approach of [13] which relies on temporal logic, and [14,15] which also uses constraint programming. Both aim at determining a class of models which are consistent with available partial data on the regulatory structure and dynamical properties. Our method is based on a model rather than on constraints, which allows to define some properties on the system structure (such as cooperations). Furthermore, we claim that we are able to deal with larger biological networks.

Outline. Sect. 2 recalls the PH and Thomas frameworks; Sect. 3 defines the IG inference from PH; Sect. 4 details the enumeration of Thomas parametrizations compatible with a PH and discuss its implementation in ASP. Sect. 5 illustrates the applicability of our method on simple examples and large biological models.

Notations. $[i;j]$ is the set of integers $\{i, i+1, \ldots, j\}$; we note $[i_1;j_1] \leq_{[]} [i_2;j_2] \overset{\Delta}{\Leftrightarrow} (i_1 \leq i_2 \wedge j_1 \leq j_2)$ and $[i_1;j_1] <_{[]} [i_2;j_2] \overset{\Delta}{\Leftrightarrow} (i_1 < i_2 \wedge j_1 \leq j_2) \vee (i_1 \leq i_2 \wedge j_1 < j_2)$. Given an integer k, $k < [i;j] \overset{\Delta}{\Leftrightarrow} k < i$ and $k > [i;j] \overset{\Delta}{\Leftrightarrow} k > j$.

2 Frameworks

2.1 The Process Hitting framework

We recall here the definition and semantics of the Process Hitting (PH), and its usage to model cooperation between concurrent components. Two examples of PH modeling a BRN at different abstraction levels are given. They serve as running examples in the rest of this article.

A PH (Def. 1) gathers a finite number of concurrent *processes* grouped into a finite set of *sorts*. A process belongs to a unique sort and is noted a_i where a is the sort and i the identifier of the process within the sort a. At any time, one and only one process of each sort is present; a state of the PH thus corresponds to the set of such processes.

The concurrent interactions between processes are defined by a set of *actions*. Actions describe the replacement of a process by another of the same sort conditioned by the presence of at most one other process in the current state of the PH. An action is denoted by $a_i \to b_j \nearrow b_k$ where a_i, b_j, b_k are processes of sorts a and b. It is required that $b_j \neq b_k$ and that $a = b \Rightarrow a_i = b_j$. An action $h = a_i \to b_j \nearrow b_k$ is read as "a_i *hits* b_j to make it bounce to b_k", and a_i, b_j, b_k are called respectively *hitter*, *target* and *bounce* of the action, and can be referred to as $\mathsf{hitter}(h), \mathsf{target}(h), \mathsf{bounce}(h)$, respectively.

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

- $\Sigma \triangleq \{a, b, \dots\}$ *is the finite set of* sorts*;*
- $L \triangleq \prod_{a\in\Sigma} L_a$ *is the set of states with* $L_a = \{a_0, \dots, a_{l_a}\}$ *the finite set of* processes *of sort* $a \in \Sigma$ *and* l_a *a positive integer with* $a \neq b \Rightarrow \forall (a_i, b_j) \in L_a \times L_b, a_i \neq b_j$*;*
- $\mathcal{H} \triangleq \{a_i \to b_j \Rsh b_k, \cdots \mid (a,b) \in \Sigma^2 \wedge (a_i, b_j, b_k) \in L_a \times L_b \times L_b \wedge b_j \neq b_k \wedge a = b \Rightarrow a_i = b_j\}$ *is the finite set of* actions*.*

$\mathcal{P}$ *denotes the set of all processes* ($\mathcal{P} \triangleq \{a_i \mid a \in \Sigma \wedge a_i \in L_a\}$)*.*

The sort of a process a_i is referred to as $\Sigma(a_i) = a$ and the set of sorts present in an action $h \in \mathcal{H}$ as $\Sigma(h) = \{\Sigma(\mathsf{hitter}(h)), \Sigma(\mathsf{target}(h))\}$. Given a state $s \in L$, the process of sort $a \in \Sigma$ present in s is denoted by $s[a]$, that is the a-coordinate of the state s. If $a_i \in L_a$, we define the notation $a_i \in s \triangleq s[a] = a_i$.

An action $h = a_i \to b_j \Rsh b_k \in \mathcal{H}$ is *playable* in $s \in L$ if and only if $s[a] = a_i$ and $s[b] = b_j$. In such a case, $(s \cdot h)$ stands for the state resulting from the play of the action h in s, that is $(s \cdot h)[b] = b_k$ and $\forall c \in \Sigma, c \neq b, (s \cdot h)[c] = s[c]$. For the sake of clarity, $((s \cdot h) \cdot h')$, $h' \in \mathcal{H}$ is abbreviated as $(s \cdot h \cdot h')$.

Example. Fig. 1 represents a PH $(\Sigma, L, \mathcal{H})$ with $\Sigma = \{a, b, c\}$, $L_a = \{a_0, a_1, a_2\}$, $L_b = \{b_0, b_1\}$, $L_c = \{c_0, c_1\}$, and

$$\begin{aligned}
\mathcal{H} = \{&a_2 \to b_1 \Rsh b_0, && b_0 \to a_2 \Rsh a_1, && c_0 \to a_2 \Rsh a_1,\\
&&& b_0 \to a_1 \Rsh a_0, && c_0 \to a_1 \Rsh a_0,\\
&&& b_1 \to a_0 \Rsh a_1, && c_1 \to a_0 \Rsh a_1,\\
&&& b_1 \to a_1 \Rsh a_2, && c_1 \to a_1 \Rsh a_2\} \ .
\end{aligned}$$

The action $h = b_1 \to a_1 \Rsh a_2$ is playable in the state $s = \langle b_1, a_1, c_0\rangle$; and $s \cdot h = \langle b_1, a_2, c_0\rangle$.

This PH example actually models a BRN where the component a has three qualitative levels and components b and c are boolean. In this BRN, b and c activate a, while a inhibits b. The inhibition of b by a is only effective when a is at level 2; in the other cases, b cannot evolve in any direction. The activation of a by b (c) is encoded by the actions making the level of a increase (resp. decrease) when b (c) is present (resp. absent). It is worth noticing that the activation of a by b (c) is independent from c (b). This may express a lack of knowledge on the cooperation between these two regulators: we thus model an over-approximation of the possible actions.

Modeling cooperation. As described in [10], the cooperation between processes to make another bounce can be expressed in PH by building a *cooperative sort*. Fig. 2 shows an example of cooperation between processes b_1 and c_1 to make a_1 bounce to a_2: a cooperative sort bc is defined with 4 processes (one for each sub-state of the presence of processes b_1 and c_1). For the sake of clarity, the bc

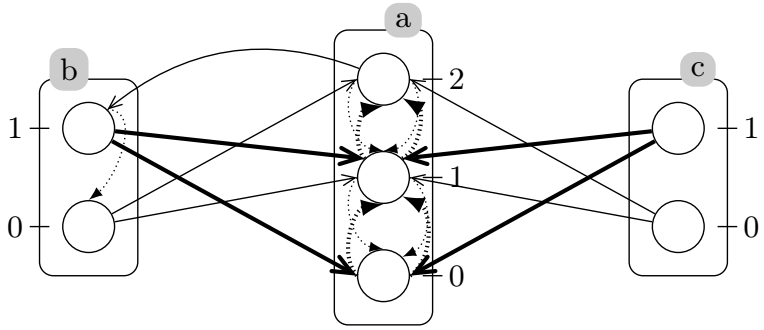


Fig. 1. A Process Hitting (PH) example. Sorts are represented by labeled boxes, and processes by circles (ticks are the identifiers of the processes within the sort, for instance, a_0 is the process ticked 0 in the box a). An action (for instance $b_1 \to a_1 \Rsh a_2$) is represented by a pair of directed arcs, having the hit part (b_1 to a_1) in plain line and the bounce part (a_1 to a_2) in dotted line. Actions involving b_1 or c_1 are in thick lines.

processes are indexed using the sub-state they represent. Hence, bc_{01} represents the sub-state $\langle b_0, c_1 \rangle$, and so on. Each process of sort b and c hit bc to make it bounce to the process reflecting the status of the sorts b and c (e.g., $b_1 \to bc_{00} \Rsh bc_{10}$ and $b_1 \to bc_{01} \Rsh bc_{11}$). Then, it is the process bc_{11} which hits a_1 to make it bounce to a_2 instead of the independent hits from b_1 and c_1.

We note that cooperative sorts are standard PH sorts and do not involve any special treatment regarding the semantics of related actions.

When the number of cooperating processes is large, it is possible to chain several cooperative sorts to prevent the combinatoric explosion of the number of processes created within cooperative sorts. For instance, if b_1, c_1, and d_1 cooperate, one can create a cooperative sort bc with 4 processes reflecting the presence of b_1 and c_1, and a cooperative sort bcd with 4 processes reflecting the presence of bc_{11} and d_1. Such constructions are helpful in PH as the static analysis of dynamics developed in [11] does not suffer from the number of sorts, but on the number of processes within a single sort.

While the construction of cooperation in PH allows to encode any boolean functions between cooperating processes [10], it is worth noticing they introduce a temporal shift in their application. This allows the existence of interleaving of actions leading to a cooperative sort representing a past sub-state of the presence of the cooperative processes. The resulting behavior is then an over-approximation of the realization of an instantaneous cooperation.

Example. The PH in Fig. 3 results from the refinement of the PH in Fig. 1 where several cooperations have been specified. In particular, the bounce to a_2 is the result of a cooperation between b_1 and c_1; and the bounce to a_0 of a cooperation between b_0 and c_0. Hence, this PH expresses a BRN where a requires both b and

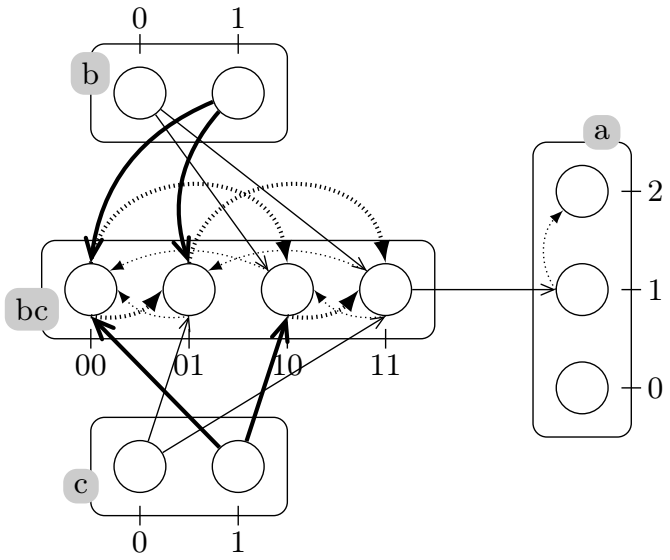


Fig. 2. A PH modeling a cooperativity between b_1 and c_1 to make a_1 bounce to a_2. Actions involving b_1 or c_1 are in thick lines.

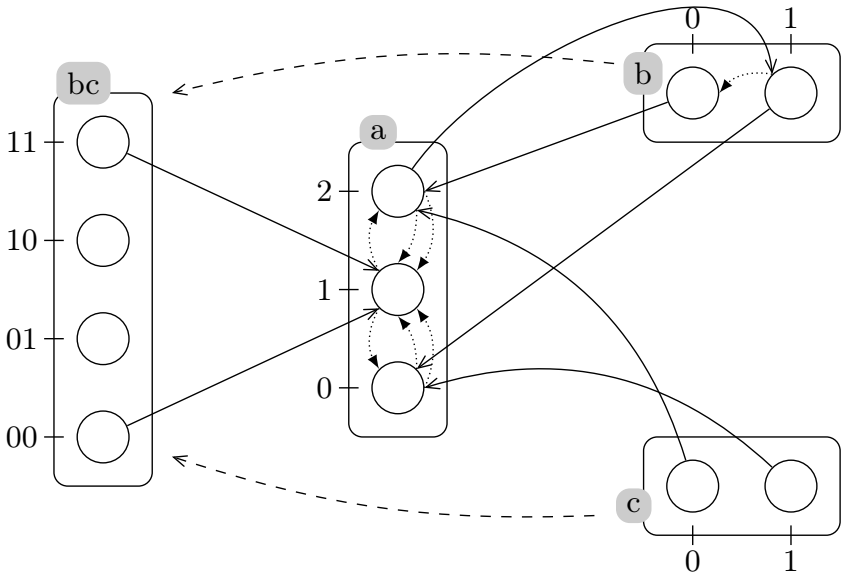


Fig. 3. PH resulting from the refinement of the one in Fig. 1 by the specification of several cooperations. The actions from b and c to the cooperative sort bc are identical to those defined in Fig. 2 and are represented here by a single dashed arc.

c active to reach its highest level, and a does not become inactive unless both b and c are inactive.

2.2 Thomas' modeling

We concisely present the Thomas' modeling of BRNs dynamics, merely inspired by [5,16]. In order to enlarge the class of Thomas' models compatible with PH dynamics (w.r.t. the presented inference), we extend the classical formalism by setting parameters to intervals of values instead of single values, and briefly discuss this addition.

Thomas' formalism lies on two complementary descriptions of the system. First, the *Interaction Graph* (IG) models the structure of the system by defining the components' mutual influences. The *parametrization* then specifies the levels to which tends a component when a given configuration of its regulators applies.

The IG is composed of nodes that represent components, and edges labeled with a threshold that stand for either positive or negative regulations (Def. 2). For such a regulation to take place, the expression level of its head component has to be higher than its threshold; otherwise, the opposite influence is expressed. The uniqueness of these regulations makes the following sections simpler. We call $\mathsf{levels}_+(a \to b)$ (resp. $\mathsf{levels}_-(a \to b)$) the levels of a where it is an activator (resp. inhibitor) of b (Def. 3); l_a denotes the maximum level of a.

Definition 2 (Interaction Graph). *An* Interaction Graph *(IG) is a triple* (Γ, E_+, E_-) *where* Γ *is a finite number of* components*, and* E_+ *(resp.* E_-*)* $\subset \{a \xrightarrow{t} b \mid a, b \in \Gamma \wedge t \in [1; l_a]\}$ *is the set of positive (resp. negative)* regulations *between two nodes, labeled with a* threshold*.*

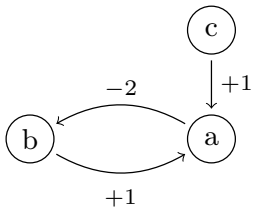
A regulation from a *to* b *is uniquely referenced: if* $a \xrightarrow{t} b \in E_+$ *(resp.* E_-*),* $\nexists a \xrightarrow{t'} b \in E_+$ *(resp.* E_-*),* $t' \neq t$ *and* $\nexists a \xrightarrow{t'} b \in E_-$ *(resp.* E_+*),* $t' \in \mathbb{N}$*.*

Definition 3 (Effective levels (levels)). *Let* (Γ, E_+, E_-) *be an IG and* $a, b \in \Gamma$ *two of its components:*

- *if* $a \xrightarrow{t} b \in E_+$*,* $\mathsf{levels}_+(a \to b) \triangleq [t; l_a]$ *and* $\mathsf{levels}_-(a \to b) \triangleq [0; t-1]$*;*
- *if* $a \xrightarrow{t} b \in E_-$*,* $\mathsf{levels}_+(a \to b) \triangleq [0; t-1]$ *and* $\mathsf{levels}_-(a \to b) \triangleq [t; l_a]$*;*
- *otherwise,* $\mathsf{levels}_+(a \to b) \triangleq \mathsf{levels}_-(a \to b) \triangleq \emptyset$*.*

For all component $a \in \Gamma$, $\Gamma^{-1}(a) \triangleq \{b \in \Gamma \mid \exists b \xrightarrow{t} a \in E_+ \cup E_-\}$ is the set of its regulators. We allow any number of levels for the components, without considering the number of outgoing edges, as the number of processes in a PH sort is not constrained in any way.

Example. Fig. 4(left) represents an Interaction Graph (Γ, E_+, E_-) with $\Gamma = \{a, b, c\}$, $E_+ = \{b \xrightarrow{1} a, c \xrightarrow{1} a\}$ and $E_- = \{a \xrightarrow{2} b\}$; hence $\Gamma^{-1}(a) = \{b, c\}$.



$K_{a,\{b,c\},\emptyset} = [2;2]$ $\quad$ $K_{b,\{a\},\emptyset} = [0;1]$

$K_{a,\{b\},\{c\}} = [1;1]$ $\quad$ $K_{b,\emptyset,\{a\}} = [0;0]$

$K_{a,\{c\},\{b\}} = [1;1]$

$K_{a,\emptyset,\{b,c\}} = [0;0]$ $\quad$ $K_{c,\emptyset,\emptyset} = [0;1]$

Fig. 4. (left) IG example. Regulations are represented by the edges labeled with their sign and threshold. For instance, the edge from b to a is labeled $+1$, which stands for: $b \xrightarrow{1} a \in E_+$. (right) Example parametrization of the left IG.

A *state* s of an IG (Γ, E_+, E_-) is an element in $\prod_{a\in\Gamma}[0; l_a]$. $s[a]$ refers to the level of component a in s. The specificity of Thomas' approach lies in the use of discrete *parameters* to represent the focal level interval towards which the component will evolve in each configuration of its regulators (Def. 4). Indeed, for each possible state of a BRN, all regulators of a component a can be divided into *activators* and *inhibitors*, given their type of interaction and expression level, referred to as the *resources* of a in this state (Def. 5).

Definition 4 (Discrete parameter $K_{a,A,B}$ and Parametrization K). *For a given component $a \in \Gamma$ and A (resp. B) $\subset \Gamma^{-1}(a)$ a set of its activators (resp. inhibitors) such that $A \cup B = \Gamma^{-1}(a)$ and $A \cap B = \emptyset$, the discrete* parameter *$K_{a,A,B} = [i; j]$ is a non-empty interval towards which a will tend in the states where its activators (resp. inhibitors) are the regulators in set A (resp. B). The complete map K of discrete parameters for $\mathcal{G}$ is called a* parametrization *of $\mathcal{G}$.*

Definition 5 (Resources $\mathsf{Res}_a(s)$). *For a given state s of a BRN, we define the* activators *A and* inhibitors *B of a in s as $\mathsf{Res}_a(s) = A, B$, where:*

$$A = \{b \in \Gamma \mid s[b] \in \mathsf{levels}_+(b \to a)\}$$
$$B = \{b \in \Gamma \mid s[b] \in \mathsf{levels}_-(b \to a)\}$$

We also denote: $\mathsf{Res}_a = \{(A; B) \mid \exists s \in \prod_{a\in\Gamma}[0; l_a], \mathsf{Res}_a(s) = A, B\}$

At last, Def. 6 gives the asynchronous dynamics of a BRN using Thomas' parameters. From a given state s, a transition to another state s' is possible provided that only one component a will evolve of one level towards $K_{a,\mathsf{Res}_a(s)}$.

Definition 6 (Asynchronous dynamics). *Let s be a state of a BRN using Thomas' parameters $(\mathcal{G}, K)$ where $\mathcal{G} = (\Gamma, E_+, E_-)$. The state that succeeds to s is given by the indeterministic function $f(s)$:*

$$f(s) = s' \Leftrightarrow \exists a \in \Gamma, s'[a] = f^a(s) \wedge \forall b \in \Gamma, b \neq a, s[b] = s'[b] \quad \text{, with}$$

$$f^a(s) = \begin{cases} s[a] + 1 & \text{if } s[a] < K_{a,A,B} \\ s[a] & \text{if } s[a] \in K_{a,A,B} \\ s[a] - 1 & \text{if } s[a] > K_{a,A,B} \end{cases} \text{, where } A, B = \mathsf{Res}_a(s).$$

While the use of intervals as parameter values does not add expressivity in boolean networks, it allows to specify a larger range of dynamics in the general case (w.r.t. the above definitions). Indeed, assume that $K_{a,A,B} = [i; i+2]$; we aim at obtaining three different parameters $K_{a,A_1,B_1} = i$, $K_{a,A_2,B_2} = i+1$, $K_{a,A_3,B_3} = i+2$. The only possible modification in resources is to add a as a self-regulator. However, because resources have a boolean definition (a component is either an activator or an inhibitor of a), it is not possible to differentiate the 3 values. We also remark that the use of intervals makes optional some explicit auto-activations in the IG (as for b in Fig. 4, for instance).

Example. In the BRN that consists of the IG and parametrization of Fig. 4, the following transitions are possible given the semantics defined in Def. 6: $\langle a_0, b_1, c_1\rangle \to \langle a_1, b_1, c_1\rangle \to \langle a_2, b_1, c_1\rangle \to \langle a_2, b_0, c_1\rangle \to \langle a_1, b_0, c_1\rangle$, ending in a steady state, where a_i is the component a at level i. As $K_{b,\{a\},\emptyset} = [0; 1]$, no auto-regulation on b is needed to prevent its evolution when a is not at level 2.

3 Interaction Graph Inference

In order to infer a complete BRN, one has to find the Interaction Graph (IG) first, as some constraints on the parametrization rely on it. Inferring the IG is an abstraction step which consists in determining the global influence of components on each of its successors.

This section first introduces the notion of focal processes within a PH (Subsect. 3.1) which is used to characterize well-formed PH for IG inference in Subsect. 3.2, and as well used by the parametrization inference presented in Sect. 4. Finally, the rules for inferring the interactions between components from a PH are described in Subsect. 3.3. We consider hereafter a global PH $(\Sigma, L, \mathcal{H})$ on which the IG inference is to be performed.

3.1 Focal Processes

Many of the inferences defined in the rest of this paper rely on the knowledge of *focal processes* w.r.t. a given context (a set of processes that are potentially present). When such a context applies, we expect to (always) reach one focal process in a bounded number of actions.

For $S_a \subseteq L_a$ and a context (set of processes) ς, let us define as $\mathcal{H}(S_a, \varsigma)$ the set of actions on the sort a having their hitter in ς and target in S_a (Eq. (1)); and the digraph (V, E) where arcs are the bounces within the sort a triggered by actions in $\mathcal{H}(S_a, \varsigma)$ (Eq. (2)). $\mathsf{focals}(a, S_a, \varsigma)$ denotes the set of focal processes of sort a in the scope of $\mathcal{H}(S_a, \varsigma)$ (Def. 7).

$$\mathcal{H}(S_a, \varsigma) \triangleq \{b_i \to a_j \Rsh a_k \in \mathcal{H} \mid b_i \in \varsigma \wedge a_j \in S_a\} \tag{1}$$

$$\begin{aligned} E &\triangleq \{(a_j, a_k) \in (S_a \times L_a) \mid \exists b_i \to a_j \Rsh a_k \in \mathcal{H}(S_a, \varsigma)\} \\ V &\triangleq S_a \cup \{a_k \in L_a \mid \exists (a_j, a_k) \in E\} \end{aligned} \tag{2}$$

Definition 7 ($\mathsf{focals}(a, S_a, \varsigma)$)**.** *The set of processes that are focal for processes in S_a in the scope of $\mathcal{H}(S_a, \varsigma)$ are given by:*

$$\mathsf{focals}(a, S_a, \varsigma) \triangleq \begin{cases} \{a_i \in V \mid \nexists (a_i, a_j) \in E\} & \text{if the digraph } (V, E) \text{ is acyclic,} \\ \emptyset & \text{otherwise.} \end{cases}$$

We note $L(\varsigma)$ the set of states $s \in L$ such that $\forall a \in \Sigma(\varsigma), s[a] \in \varsigma$, where $\Sigma(\varsigma)$ is the set of sorts with processes in ς. We say a sequence of actions $h^1, \ldots, h^n$ is *bounce-wise* if and only if $\forall m \in [1; n-1], \mathsf{bounce}(h^m) = \mathsf{target}(h^{m+1})$. From Def. 7, it derives that:

1. if $\mathsf{focals}(a, S_a, \varsigma) = \emptyset$, there exists a state $s \in L(\varsigma \cup S_a)$ such that $\forall n \in \mathbb{N}$ there exists a bounce-wise sequence of actions $h^1, \ldots, h^{n+1}$ in $\mathcal{H}(S_a, \varsigma)$ with $\mathsf{target}(h^1) \in s$.
2. if $\mathsf{focals}(a, S_a, \varsigma) \neq \emptyset$, for all state $s \in L(\varsigma \cup S_a)$, for any bounce-wise sequence of actions $h^1, \ldots, h^n$ in $\mathcal{H}(S_a, \varsigma)$ where $\mathsf{target}(h^1) \in s$, either $\mathsf{bounce}(h^n) \in \mathsf{focals}(a, S_a, \varsigma)$, or $\exists h^{n+1} \in \mathcal{H}(a, \varsigma)$ such that $\mathsf{bounce}(h^n) = \mathsf{target}(h^{n+1})$. Moreover $n \leq |\mathcal{H}(S_a, \varsigma)|$ (i.e. no cycle of actions possible).

It is worth noticing that those bounce-wise sequences of actions may not be successively playable in a state $s \in L(\varsigma \cup S_a)$. Indeed, nothing impose that the hitters of actions are present in s. In the general case, the playability of those bounce-wise sequences, referred to as *focals reachability* may be hard to prove. However, in the scope of this paper, the particular contexts used with focals ensure this property. Notably, the rest of this section uses only *strict* contexts (Def. 8) which allow at most one hitter per sort in the bounce-wise sequences (and thus are present in s).

Definition 8 (Strict context for S_a). *A context (set of processes) ς is strict for $S_a \subseteq L_a$ if and only if $\{b_i, b_j\} \subset \varsigma \wedge b \neq a \Rightarrow i = j$.*

In other words, assuming focals reachability, if $\mathsf{focals}(a, S_a, \varsigma)$ is empty, there exists a sequence of actions that may be played an unbound number of times (cycle); if it is non-empty, it is ensured that any state in $L(\varsigma \cup S_a)$ converges, in a bounded number of steps, either to a process in S_a that is not hit by processes in ς, or to a process in $L_a \setminus S_a$.

Example. In the PH of Fig. 1, we obtain:

$$\mathsf{focals}(a, L_a, \{b_0, c_0\}) = \{a_0\} \qquad \mathsf{focals}(a, L_a, \{b_1, c_1\}) = \{a_2\}$$
$$\mathsf{focals}(a, L_a, \{b_1, c_0\}) = \emptyset \qquad \mathsf{focals}(a, \{a_1\}, \{b_1, c_0\}) = \{a_0, a_2\}$$

3.2 Well-formed Process Hitting for Interaction Graph Inference

The inference of an IG from a PH assumes that the PH defines two types of sorts: the sorts corresponding to BRN components, and the cooperative sorts. This leads to the characterization of a *well-formed* PH for IG inference.

The identification of sorts modeling components relies on the observation that their processes represent (ordered) qualitative levels. Hence an action on such a sort cannot make it bounce to a process at a distance more than one. The set of sorts satisfying such a condition is referred to as Γ (Eq. (3)). Therefore, in the rest of this paper, Γ denotes the set of components of the BRN to infer.

$$\Gamma \triangleq \{a \in \Sigma \mid \nexists b_i \rightarrow a_j \Rsh a_k \in \mathcal{H}, |j-k| > 1\} \tag{3}$$

Any sort that does not act as a component should then be treated as a cooperative sort. As explained in Subsect. 2.1, the role of a cooperative sort υ is to compute the current state of set of cooperating processes. Hence, for each sub-state σ formed by the sorts hitting υ, υ should converge to a focal process. This is expressed by Property 1, where the set of sorts having an action on a given sort a is given by $\Sigma^{-1}(a)$ (Eq. (4)) and $\mathcal{P}(\sigma)$ is the set of processes that compose the sub-state σ.

$$\forall a \in \Sigma, \Sigma^{-1}(a) \triangleq \{b \in \Sigma \mid \exists b_i \rightarrow a_j \Rsh a_k \in \mathcal{H}\} \tag{4}$$

Property 1 (Well-formed cooperative sort). A sort $\upsilon \in \Sigma$ is a well-formed cooperative sort if and only if each configuration σ of its predecessors leads υ to a unique focal process, denoted by $\upsilon(\sigma)$:

$$\forall \sigma \in \prod_{a \in \Sigma^{-1}(\upsilon) \wedge a \neq \upsilon} L_a, \mathsf{focals}(\upsilon, L_\upsilon, \mathcal{P}(\sigma) \cup L_\upsilon) = \{\upsilon(\sigma)\}$$

Such a property allows a large variety of definitions of a cooperative sort, but for the sake of simplicity, does not allow the existence of multiple focal processes. While this may be easily extended to (the condition becomes $\mathsf{focals}(\upsilon, L_\upsilon, \mathcal{P}(\sigma) \cup L_\upsilon) \neq \emptyset$), it makes some hereafter equations a bit more complex to read as they should handle a set of focal processes instead of a unique focal process.

Finally, Property 2 sums up the conditions for a Process Hitting to be suitable for IG inference. In addition of having either component sorts or well-formed cooperative sorts, we also require that there is no cycle between cooperative sorts, and that sorts being never hit (i.e. serving as an invariant environment) are components.

Property 2 (Well-formed Process Hitting for IG inference). A PH is well-formed for IG inference if and only if the following conditions are verified:

- each sort $a \in \Sigma$ either belongs to Γ, or is a well-formed cooperative sort;
- there is no cycle between cooperative sorts (the digraph $(\Sigma, \{(a,b) \in (\Sigma \times \Sigma) \mid \exists a_i \rightarrow b_j \Rsh b_k \in \mathcal{H} \wedge a \neq b \wedge \{a,b\} \cap \Gamma = \emptyset\})$ is acyclic);
- sorts having no action hitting them belong to Γ ($\{a \in \Sigma \mid \nexists b_i \rightarrow a_j \Rsh a_k \in \mathcal{H}\} \subset \Gamma$).

Example. In the PH of Fig. 3, bc is a well-formed cooperative sort as defined in Property 1, because:

$$\mathsf{focals}(bc, L_{bc}, \{b_0, c_0\} \cup L_{bc}) = \{bc_{00}\} \quad \mathsf{focals}(bc, L_{bc}, \{b_0, c_1\} \cup L_{bc}) = \{bc_{01}\}$$
$$\mathsf{focals}(bc, L_{bc}, \{b_1, c_0\} \cup L_{bc}) = \{bc_{10}\} \quad \mathsf{focals}(bc, L_{bc}, \{b_1, c_1\} \cup L_{bc}) = \{bc_{11}\}$$

Hence, both Fig. 1 and Fig. 3 are well-formed PH for IG inference with $\Gamma = \{a, b, c\}$.

3.3 Interaction Inference

At this point we can divide the set of sorts Σ into components (Γ, see Eq. (3)) and cooperative sorts ($\Sigma \setminus \Gamma$) that will not appear in the IG. We define in Eq. (5) the set of predecessors of a sort a, that is, the sorts influencing a by considering direct actions and possible intermediate cooperative sorts. The predecessors of a that are components are the regulators of a, denoted $\mathsf{reg}(a)$ (Eq. (6)).

$$\begin{aligned}\forall a \in \Sigma, \mathsf{pred}(a) \triangleq \{b \in \Sigma \mid \exists n \in \mathbb{N}^*, \exists (c^k)_{k\in[0;n]} \in \Sigma^{n+1}, \\ c^0 = b \wedge c^n = a \\ \wedge \forall k \in [\![0; n-1]\!], c^k \in \Sigma^{-1}(c^{k+1}) \cap (\Sigma \setminus \Gamma)\}\end{aligned} \tag{5}$$

$$\forall a \in \Sigma, \mathsf{reg}(a) \triangleq \mathsf{pred}(a) \cap \Gamma \tag{6}$$

Given a set g of components and a configuration (i.e. a sub-state) σ, $\varsigma_g(\sigma)$ refers to the set of processes hitting a regulated by any sort in g (Eq. (7)). If $g = \{b\}$, we simple note $\varsigma_b(\sigma)$. This set is composed of the active processes of sorts in g, and the focal process (assumed unique) of the cooperative sorts υ hitting a that have a predecessor in g. The evaluation of the focal process of υ in context σ, denoted $\upsilon(\sigma)$, relies on Property 1, which gives its value when all the direct predecessors of υ are defined in σ. When a predecessor υ' is not in σ, we extend the evaluation by recursively computing the focal value of υ' is σ, as stated in Eq. (8). Because there is no cycle between cooperative sorts, this recursive evaluation of $\upsilon(\sigma)$ always terminates.

$$\forall g \subset \Gamma, \varsigma_g(\sigma) \triangleq \{\sigma[b] \mid b \in g\} \cup \{\upsilon(\sigma) \mid \upsilon \in \Sigma^{-1}(a) \setminus \Gamma \wedge g \cap \mathsf{reg}(\upsilon) \neq \emptyset\} \tag{7}$$

$$\upsilon(\sigma) \triangleq \upsilon(\sigma \uplus \langle \upsilon'(\sigma) \mid \upsilon' \in \Sigma^{-1}(\upsilon) \wedge \upsilon' \in \Sigma \setminus \Gamma \rangle) \tag{8}$$

We aim at inferring that b activates (inhibits) a if there exists a configuration where increasing the level of b makes possible the increase (decrease) of the level of a. This is analogous to standard IG inferences from discrete maps [2].

This reasoning can be straightforwardly applied to a PH when inferring the influence of b on a when $b \neq a$ (Eq. (11)). Let us define $\gamma(b \to a)$ as the set of components cooperating with b to hit a, including b and a (Eq. (9)). Given a configuration $\sigma \in \prod_{c\in\gamma(b\to a)} L_c$, $\mathsf{focals}(a, \{a_i\}, \varsigma_b(\sigma))$ gives the bounces that a given process a_i can make in the context $\varsigma_b(\sigma)$. We note $\sigma\{b_i\}$ the configuration σ where the process of sort b has been replaced by b_i. If there exists $b_i, b_{i+1} \in L_b$ such that one bounce in $\mathsf{focals}(a, \{\sigma[a]\}, \varsigma_b(\sigma\{b_i\}))$ has a lower (resp. higher) level that one bounce in $\mathsf{focals}(a, \{\sigma[a]\}, \varsigma_b(\sigma\{b_{i+1}\}))$, then b as positive (resp. negative) influence on a with a maximum threshold $l = i + 1$.

$$\gamma(b \to a) \triangleq \{a, b\} \cup \{c \in \Gamma \mid \exists \upsilon \in \Sigma \setminus \Gamma, \upsilon \in \mathsf{pred}(a) \wedge \{b, c\} \subset \mathsf{pred}(\upsilon)\} \tag{9}$$

Then, we infer that a has a self-influence if its current level can have an impact on its own evolution at a given configuration σ. We consider here a configuration σ of a group g of sorts having a cooperation on a. This set of sort groups is given by $X(a)$ (Eq. (10)) which returns the set of connected components (noted $\mathcal{C}$) of the graph linking two regulators b, c of a if there is a cooperative sort hitting a regulated by both of them. Given $a_i, a_{i+1} \in L_a$, we pick $a_j \in \mathsf{focals}(a, \{a_i\}, \varsigma_g(\sigma\{a_i\}))$ and $a_k \in \mathsf{focals}(a, \{a_{i+1}\}, \varsigma_g(\sigma\{a_{i+1}\}))$. If $k = j+1$, we can not conclude as there is no difference in the evolution of both levels. If $k \neq j+1$ and $k-j \neq 0$, then a_i and a_{i+1} have divergent evolutions: we infer an influence of sign of $k-j$ at threshold $i+1$. We note that some aspects of this inference are arbitrary and may impact the number of parameters to infer in the next section. In particular, in some cases, the use of intervals for Thomas' parameters drops the requirement of inferring a self-activation.

$$X(a) = \mathcal{C}\left((\mathsf{reg}(a), \{\{b, c\} \mid \exists \upsilon \in \Sigma^{-1}(a) \setminus \Gamma, \{b, c\} \subset \mathsf{reg}(\upsilon)\})\right) \tag{10}$$

Proposition 1 details the inference of all existing influences between components occurring with a threshold l. These influences are split into positive and negative ones, and represent possible edges in the final IG. We do not consider the cases where a component has no visible influence on another.

Proposition 1 (Edges inference). *We define the set of positive (resp. negative) influences $\hat{E}_+$ (resp. $\hat{E}_-$) for any $a \in \Gamma$ by:*

$$\begin{aligned}
&\forall b \in \mathsf{reg}(a), b \neq a,\\
&\quad b \xrightarrow{l} a \in \hat{E}_s \iff \exists \sigma \in \textstyle\prod_{c \in \gamma(b \to a)} L_c, \exists b_i, b_{i+1} \in L_b,\\
&\qquad\qquad \exists a_j \in \mathsf{focals}(a, \{\sigma[a]\}, \varsigma_b(\sigma\{b_i\})),\\
&\qquad\qquad \exists a_k \in \mathsf{focals}(a, \{\sigma[a]\}, \varsigma_b(\sigma\{b_{i+1}\})),\\
&\qquad\qquad\quad s = \mathsf{sign}(k-j) \wedge l = i+1
\end{aligned} \tag{11}$$

$$\begin{aligned}
&a \xrightarrow{l} a \in \hat{E}_s \iff \exists g \in X(a), \sigma \in L_a \times \textstyle\prod_{b \in g} L_b, \exists a_i, a_{i+1} \in L_a,\\
&\qquad\qquad \exists a_j \in \mathsf{focals}(a, \{a_i\}, \varsigma_g(\sigma\{a_i\})),\\
&\qquad\qquad \exists a_k \in \mathsf{focals}(a, \{a_{i+1}\}, \varsigma_g(\sigma\{a_{i+1}\})),\\
&\qquad\qquad\quad k \neq j+1 \wedge s = \mathsf{sign}(k-j) \wedge l = i+1
\end{aligned} \tag{12}$$

where $s \in \{+, -\}$, $\bar{s} = + \overset{\Delta}{\Leftrightarrow} s = -$, $\bar{s} = - \overset{\Delta}{\Leftrightarrow} s = +$, $\mathsf{sign}(n) = + \overset{\Delta}{\Leftrightarrow} n > 0$, $\mathsf{sign}(n) = - \overset{\Delta}{\Leftrightarrow} n < 0$, *and* $\mathsf{sign}(0) \overset{\Delta}{=} 0$.

We are now able to infer the edges of the final IG by considering positive and negative influences (Proposition 2). We infer a positive (resp. negative) edge if there exists a corresponding influence with the same sign. If an influence is both positive and negative, we infer an unsigned edge. In the end, the threshold of each edge is the minimum threshold for which the influence has been found. As unsigned edges represent ambiguous interactions, no threshold is inferred.

Proposition 2 (Interaction Graph inference). *We infer* $\mathcal{G} = (\Gamma, E_+, E_-, E_\pm)$ *from Proposition 1 as follows:*

$$E_+ = \{a \xrightarrow{t} b \mid \nexists a \xrightarrow{t'} b \in \hat{E}_- \wedge t = \min\{l \mid a \xrightarrow{l} b \in \hat{E}_+\}\}$$
$$E_- = \{a \xrightarrow{t} b \mid \nexists a \xrightarrow{t'} b \in \hat{E}_+ \wedge t = \min\{l \mid a \xrightarrow{l} b \in \hat{E}_-\}\}$$
$$E_\pm = \{a \to b \mid \exists a \xrightarrow{t} b \in \hat{E}_+ \wedge \exists a \xrightarrow{t'} b \in \hat{E}_-\}$$

Example. The IG inference from the PH of Fig. 3 gives $\hat{E}_+ = \{b \xrightarrow{1} a, c \xrightarrow{1} a\}$ and $\hat{E}_- = \{a \xrightarrow{2} b\}$, corresponding to the IG of Fig. 4. No self-influence are inferred ($X(a) = \{\{b, c\}\}$, $X(b) = \{\{a\}\}$, and $X(c) = \emptyset$).

4 Parametrization inference

Given the IG inferred from a PH as presented in the previous section, one can find the discrete parameters that model the behavior of the studied PH using the method presented in the following. It relies on an exhaustive enumeration of all predecessors of each component in order to find attractor processes and returns a possibly incomplete parametrization, given the exhaustiveness of the cooperations. The last step consists of the enumeration of all compatible complete parametrizations given this set of inferred parameters, the PH dynamics and some biological constraints on parameters.

4.1 Parameters inference

This subsection presents some results related to the inference of independent discrete parameters from a given PH. These results are equivalent to those presented in [10], with notation adapted to be shared with the previous section. In addition, we introduce the well-formed PH for parameter inference property (Property 3), which implies that the inferred IG does not contain any unsigned interactions, and thus can be seen as the regular IG (Γ, E_+, E_-), and that any processes in $\mathsf{levels}_+(b \to a)$ (resp. $\mathsf{levels}_-(b \to a)$) share the same behavior regarding a.

Property 3 (Well-formed PH for parameter inference). A PH is well-formed for parameter inference if and only if it is well-formed for IG inference, and the IG $(\Gamma, E_+, E_-, E_\pm)$ inferred by Proposition 2 verifies $E_\pm = \emptyset$ and if the following property holds:

$$\begin{aligned} \forall b \in \Gamma^{-1}(a), \forall (i, j \in \mathsf{levels}_+(b \to a) \vee i, j \in \mathsf{levels}_-(b \to a)), \\ \forall c, ((b \neq a \wedge c = a) \vee (c \in \mathsf{pred}(a) \wedge b \in \Sigma^{-1}(c))), \\ b_i \to c_k \Rsh c_l \in \mathcal{H} \Leftrightarrow b_j \to c_k \Rsh c_l \in \mathcal{H} \end{aligned} \tag{13}$$

Let $K_{a,A,B}$ be the parameter we want to infer for a given component $a \in \Gamma$ and $A \subset \Gamma^{-1}(a)$ (resp. $B \subset \Gamma^{-1}(a)$) a set of its activators (resp. inhibitors). This inference, as for the IG inference, relies on the search of focal processes of the component for the given configuration of its regulators.

For each sort $b \in \Gamma^{-1}(a)$, we define a context $C^b_{a,A,B}$ in Eq. (14) that contains all processes representing the influence of the regulators in the configuration A, B. The context of a cooperative sort υ that regulates a is given in Eq. (15) as the set of focal processes matching the current configuration. $C_{a,A,B}$ refers to the union of all these contexts (Eq. (16)).

$$\forall b \in \Gamma,\ C^b_{a,A,B} \triangleq \begin{cases} \mathsf{levels}_+(b \to a) & \text{if } b \in A, \\ \mathsf{levels}_-(b \to a) & \text{if } b \in B, \\ L_b & \text{otherwise;} \end{cases} \tag{14}$$

$$\forall \upsilon \in \mathsf{pred}(a) \setminus \Gamma,\ C^\upsilon_{a,A,B} \triangleq \{\upsilon(\sigma) \mid \sigma \in \textstyle\prod_{c \in \Sigma^{-1}(\upsilon)} C^c_{a,A,B}\} \tag{15}$$

$$C_{a,A,B} \triangleq \textstyle\bigcup_{b \in \mathsf{pred}(a)} C^b_{a,A,B} \tag{16}$$

The parameter $K_{a,A,B}$ specifies to which values a eventually evolves as long as the context $C_{a,A,B}$ holds, which is precisely the definition of the focals function (Def. 7 in Subsect. 3.1), where the focals reachability property can be derived from Property 3 and Eq. (15). Hence $K_{a,A,B} = \mathsf{focals}(a, C^a_{a,A,B}, C_{a,A,B})$ if this latter is a non-empty interval (Proposition 3).

Proposition 3 (Parameter inference). *Let $(\Sigma, L, \mathcal{H})$ be a Process Hitting well-formed for parameter inference, and $\mathcal{G} = (\Gamma, E_+, E_-)$ the inferred IG. Let A (resp. B) $\subseteq \Gamma$ be the set of regulators that activate (resp. inhibit) a sort a. If $\mathsf{focals}(a, C^a_{a,A,B}, C_{a,A,B}) = [a_i; a_j]$ is a non-empty interval, then $K_{a,A,B} = [i; j]$.*

Example. Applied to the PH in Fig. 1, we obtain, for instance, $K_{b,\{a\},\emptyset} = [0; 1]$ and $K_{a,\{b,c\},\emptyset} = [2; 2]$, while $K_{a,\{b\},\{c\}}$ can not be inferred. For the PH in Fig. 3, this latter is evaluated to $[1; 1]$.

Given the Proposition 3, we see that in some cases, the inference of the targeted parameter is impossible. This can be due to a lack of cooperation between regulators: when two regulators independently hit a component, their actions can have opposite effects, leading to either an indeterministic evolution or to oscillations. Such an indeterminism is not possible in a BRN as in a given configuration of regulators, a component can only have an interval attractor, and eventually reaches a steady-state. In order to avoid such inconclusive cases, one has to ensure that no such behavior is allowed by either removing undesired actions or using cooperative sorts to prevent opposite influences between regulators.

4.2 Admissible parametrizations

When building a BRN, one has to find the parametrization that best describes the desired behavior of the studied system. Complexity is inherent to this process

as the number of possible parametrizations for a given IG is exponential w.r.t. the number of components. However, the method of parameters inference presented in this section gives some information about necessary parameters given a certain dynamics described by a PH. This information thus drops the number of possible parametrizations, allowing to find the desired behavior more easily.

We first delimit the validity of a parameter (Property 4) in order to ensure that any transition in the resulting BRN is allowed by the studied PH. This is verified by the existence of a hit making the concerned component bounce into the direction of the value of the parameter in the matching context. Thus, assuming Property 3 holds, any transition in the inferred BRN corresponds to at least one transition in the PH, proving the correctness of our inference. We remark that any parameter inferred by Proposition 3 satisfies this property.

Property 4 (Parameter validity). A parameter $K_{a,A,B}$ is valid w.r.t. the PH iff the following equation is verified:

$$\forall a_i \in C^a_{a,A,B}, a_i \notin K_{a,A,B} \Longrightarrow (\exists c_k \to a_i \Rsh a_j \in \mathcal{H}, c_k \in C^c_{a,A,B} \\ \wedge a_i < K_{a,A,B} \Rightarrow j > i \wedge a_i > K_{a,A,B} \Rightarrow j < i)$$

Then, we use some additional biological constraints on Thomas' parameters given in [16], that we sum up in the following three properties:

Property 5 (Extreme values assumption). Let $\mathcal{G} = (\Gamma, E_+, E_-)$ be an IG. A parametrization K on $\mathcal{G}$ satisfies the *extreme values assumption* iff:

$$\forall b \in \Gamma, \Gamma^{-1}(b) \neq \emptyset \Rightarrow 0 \in K_{b,\emptyset,\Gamma^{-1}(b)} \wedge l_b \in K_{b,\Gamma^{-1}(b),\emptyset}$$

Property 6 (Activity assumption). Let $\mathcal{G} = (\Gamma, E_+, E_-)$ be an IG. A parametrization K on $\mathcal{G}$ satisfies the *activity assumption* iff:

$$\forall b \in \Gamma, \forall a \in \Gamma^{-1}(b), \exists (A;B) \in \mathsf{Res}_a, K_{b,A,B} <_{[]} K_{b,A\cup\{b\},B\setminus\{b\}}$$
$$\forall b \in \Gamma, \forall a \in \Gamma^{-1}(b), \exists (A;B) \in \mathsf{Res}_a, K_{b,A\setminus\{b\},B\cup\{b\}} <_{[]} K_{b,A,B}$$

Property 7 (Monotonicity assumption). Let $\mathcal{G} = (\Gamma, E_+, E_-)$ be an IG. A parametrization K on $\mathcal{G}$ satisfies the *monotonicity assumption* iff:

$$\forall b \in \Gamma, \forall (A;B), (A';B') \in \mathsf{Res}_b, A \subset A' \wedge B' \subset B \Rightarrow K_{b,A,B} \leq_{[]} K_{b,A',B'}$$

4.3 Answer Set Programming implementation concepts

Answer Set Programming (ASP) [12] has been chosen to address the enumeration of all admissible parametrizations. The motivations are following:

- ASP efficiently tackles the inherent complexity of the models we use, thus allowing a fast execution of the formal tools defined in this paper,
- it is convenient to enumerate a large set of possible answers,
- it allows us to easily constrain the answers according to some properties.

We now synthesize some key points to better make the reader understand our ASP implementation with the enumeration example.

All information describing the studied model (PH and inferred IG & parameters) are expressed in ASP using facts. For functional purposes, we assign a unique label to each couple A, B of activators and inhibitors of a given component, and in the following we note $K^{p}_{a,A,B}$ the parameter of component a whose regulators A, B are assigned to the label p. Then, to state the existence of a parameter $K^{p}_{a,A,B}$, we use an atom named `param_label` in the following fact:

```
param_label(a, p).
```

Defining a set in ASP is equivalent to defining the rule for belonging to this set. For example, we define an atom `param_act` that describes the set of active regulators of a given a parameter. Describing the activators of $K^{p}_{a,\{b,c\},\{d\}}$ gives:

```
param_act(a, p, b).
param_act(a, p, c).
```

The absence of such a fact involving `d` with label `p` indicates that `d` is an inhibitor in the configuration of regulators related to this parameter.

Rules allow more detailed declarations than facts as they have a body (right-hand part below) containing constraints and allowing to use variables, while facts only have a head (left-hand part). For instance, in order to define the set of expression levels of a component, we declare:

```
component_levels(X, 0..M) :- component(X, M).
```

where the `component(X, M)` atom stands for the existence of a component `X` with a maximum level `M`. Considering this declaration, any possible answer for the atom `component_levels` will be found by binding all possible values of its two terms with all existing `component` facts: the existence of an answer `component_levels(a, k)` will depend on the existence of a term `a`, which is bound with `X`, and an integer `k`, constrained by: $0 \leq k \leq M$.

Cardinalities are convenient to enumerate all possible parametrizations by creating multiple answer sets. A cardinality (denoted hereafter with curly brackets) gives any number of possible answers for some atoms between a lower and upper bounds. For example,

```
1 { param(X, P, I) : component_levels(X, I) } :-
      param_label(X, P), not infered_param(X, P).
```

where `param(X, P, I)` stands for: `I` $\in K^{P}_{X,A,B}$, means that any parameter of component `X` and label `P` must contain at least one level value (`I`) in the possible expression levels of `X`. Indeed, the lower bound is 1, forcing at least one element in the parameter, but no upper bound is specified, allowing up to any number of answers. The body (right-hand side) of the rule also checks for the existence of a parameter of `X` with label `P`, and constrains that the parametrization inference was not conclusive for the considered parameter (`not` stands for negation by failure: `not L` becomes true if `L` is not true). Such a constraint gives multiple

results as any set of `param` atoms satisfying the cardinality will lead to a new global set of answers. In this way, we enumerate all possible parametrizations which respects the results of parameters inference, but completely disregarding the notion of admissible parametrizations given in Subsect. 4.2.

We rely on integrity constraints to filter only admissible parametrizations. An integrity constraint is a rule with no head, that makes an answer set unsatisfiable if its body turns out to be true. Hence, if we suppose that:

- the `less_active(a, p, q)` atom means that $K^{\mathtt{p}}_{a,A,B}$ stands for a configuration with less activating regulators than $K^{\mathtt{q}}_{a,A',B'}$ (i.e. $A \subset A'$),
- the `param_inf(a, p, q)` atom means: $K^{\mathtt{p}}_{a,A,B} \leq_{[]} K^{\mathtt{q}}_{a,A',B'}$,

the monotonicity assumption is formulated as the following integrity constraint:

```
:- less_active(X, P, Q), not param_inf(X, P, Q).
```

which removes all parametrization results where parameters $K^{\mathtt{P}}_{\mathtt{X},A,B}$ and $K^{\mathtt{Q}}_{\mathtt{X},A',B'}$ exist such that $A \subset A'$ and $K^{\mathtt{Q}}_{\mathtt{X},A',B'} <_{[]} K^{\mathtt{P}}_{\mathtt{X},A,B}$, thus violating the monotonicity assumption. Of course, other assumptions can be formulated in the same way.

This subsection succinctly described how we write ASP programs to represent a model and solve all steps of Thomas' modeling inference. It finds a particularly interesting application in the enumeration of parameters: all possible parametrizations are generated in separate answer sets, and integrity constraints are formulated to remove those that do not fit the assumptions of admissible parametrizations, thus reducing the number of interesting parametrizations to be considered in the end.

5 Examples

The inference method described in this paper has been implemented as part of PINT[4], which gathers PH related tools. Our implementation mainly consists in ASP programs that are solved using Clingo[5]. The IG and parameters inference can be performed using the command `ph2thomas -i model.ph --dot ig.dot` where `model.ph` is the PH model in PINT format and `ig.dot` is an output of the inferred IG in DOT format. The (possibly partial) inferred parametrization will be returned on the standard output. The admissible parametrizations enumeration is performed when adding the `--enumerate` parameter to the command.

Applied to the example in Fig. 3 where cooperations have been defined, our method infers the IG and parametrization given in Fig. 4. Regarding the example in Fig. 1, the same IG is inferred, as well as for the parametrization except for the parameters $K_{a,\{b\},\{c\}}$ and $K_{a,\{c\},\{b\}}$ which are undefined (because of the lack of cooperativity between b and c). In such a case, this partial parametrization allows 36 admissible complete parametrizations, as two parameters with 3 potential

[4] Available at `http://process.hitting.free.fr`
[5] Available at `http://potassco.sourceforge.net`

values could not be inferred. If we constrain these latter parameters so that they contain exactly one element, we obtain only 9 admissible parametrizations.

The current implementation can successfully handle large PH models of BRNs found in the literature such as an ERBB receptor-regulated G1/S transition model from [17] which contains 20 components and 15 cooperative sorts, and a T-cells receptor model from [18] which contains 40 components and 14 cooperative sorts[6]. For each model, IG and parameters inferences are performed together in less than a second on a standard desktop computer. After removing the cooperations from these models (leaving only raw actions), the inferences allow to determine 40 parameters out of 195 for the 20 components model, and 77 out of 143 for the 40 components model. As we thus have an order of magnitude of respectively 10^{31} and 10^{73} admissible parametrizations, these models would therefore be more efficiently studied as PH than as BRNs. We note that the complexity of the method is exponential in the number of regulators of one component and linear in the number of components.

A PH model can be built based on information found in the literature about the local influences between components. The precision of this knowledge will determine the precision of the modeled activations and inhibitions, and some information is likely to help in the representation of cooperations.

6 Conclusion and Discussion

This work establishes the abstraction relationship between PH and Thomas' approaches for qualitative BRN modeling. The PH allows an abstract representation of BRNs dynamics (allowing incomplete knowledge on the cooperation between components) that can not be exactly represented in René Thomas' formalism by a single instance of BRN parametrization. This motivates the concretization of PH models into a set of compatible Thomas' models in order to benefit of the complementary advantages of these two formal frameworks.

We first propose an original inference of the Interaction Graph (IG) from a BRN having its dynamics specified in the PH framework. An IG gives a compact abstract representation of the influence of the components between each others. Then, based on a prior inference of René Thomas' parametrization for BRNs from a PH model, we delimit the set of compatible Thomas' parametrizations that are compatible with the PH dynamics, and give arguments for their correctness. A parametrization is compatible with the PH if its dynamics (in terms of possible transitions) is included in the PH dynamics. The enumeration of such parametrizations is efficiently tackled using Answer Set Programming. We illustrate the overall method on simple examples and large biological models.

Several extensions of the presented work are now to be considered. First, we now plan to explore the inference of Thomas' parameters when the inferred IG involves unsigned interactions, as some particular cases having such IGs are known to have correct Thomas' parametrization (not respecting some assumptions imposed in Subsect. 4.2, however). Second, the inference of BRN multiplexes [19]

[6] Both models are available as examples distributed with PINT.

may be of practical interest as they allow to implicitly reduce the possible parametrizations by making cooperations appear in the IG. Because of its atomicity, the PH allows to specify a range of cooperations that can not be completely captured by a single instance of BRN multiplexes, then encouraging the inference of a set of compatible ones. Finally, in order to improve the performances in the IG inference, we will consider projection operations on the PH structure to undo cooperations between components and reduce the cardinality of configurations to explore by making the interactions independent.

Ack. This work was partially supported by the Fondation Centrale Initiatives.

References

1. Thomas, R.: Boolean formalization of genetic control circuits. Journal of Theoretical Biology **42**(3) (1973) 563 – 585
2. Richard, A., Comet, J.P.: Necessary conditions for multistationarity in discrete dynamical systems. Discrete Applied Mathematics **155**(18) (2007) 2403 – 2413
3. Remy, É., Ruet, P., Thieffry, D.: Graphic requirements for multistability and attractive cycles in a boolean dynamical framework. Advances in Applied Mathematics **41**(3) (2008) 335 – 350
4. Naldi, A., Remy, E., Thieffry, D., Chaouiya, C.: A reduction of logical regulatory graphs preserving essential dynamical properties. In: Computational Methods in Systems Biology. Volume 5688 of LNCS. Springer (2009) 266–280
5. Richard, A., Comet, J.P., Bernot, G.: Formal Methods for Modeling Biological Regulatory Networks. In: Modern Formal Methods and App. (2006) 83–122
6. Naldi, A., Thieffry, D., Chaouiya, C.: Decision diagrams for the representation and analysis of logical models of genetic networks. In: Computational Methods in Systems Biology, Springer (2007) 233–247
7. Siebert, H., Bockmayr, A.: Incorporating time delays into the logical analysis of gene regulatory networks. In: Computational Methods in Systems Biology. Volume 4210 of LNCS. Springer (2006) 169–183
8. Ahmad, J., Roux, O., Bernot, G., Comet, J.P., Richard, A.: Analysing formal models of genetic regulatory networks with delays. International Journal of Bioinformatics Research and Applications (IJBRA) **4**(2) (2008)
9. Twardziok, S., Siebert, H., Heyl, A.: Stochasticity in reactions: a probabilistic boolean modeling approach. In: Computational Methods in Systems Biology, ACM (2010) 76–85
10. Paulevé, L., Magnin, M., Roux, O.: Refining dynamics of gene regulatory networks in a stochastic π-calculus framework. In: Transactions on Computational Systems Biology XIII. Springer (2011) 171–191
11. Paulevé, L., Magnin, M., Roux, O.: Static analysis of biological regulatory networks dynamics using abstract interpretation. Mathematical Structures in Computer Science **in press** (2012) Preprint: `http://loicpauleve.name/mscs.pdf`.
12. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press (2003)
13. Khalis, Z., Comet, J.P., Richard, A., Bernot, G.: The SMBioNet method for discovering models of gene regulatory networks. Genes, Genomes and Genomics **3(special issue 1)** (2009) 15–22

14. Corblin, F., Fanchon, E., Trilling, L.: Applications of a formal approach to decipher discrete genetic networks. BMC Bioinformatics **11**(1) (2010) 385
15. Corblin, F., Fanchon, E., Trilling, L., Chaouiya, C., Thieffry, D.: Automatic inference of regulatory and dynamical properties from incomplete gene interaction and expression data. In: IPCAT. Volume 7223 of LNCS., Springer (2012) 25–30
16. Bernot, G., Cassez, F., Comet, J.P., Delaplace, F., Müller, C., Roux, O.: Semantics of biological regulatory networks. Electronic Notes in Theoretical Computer Science **180**(3) (2007) 3 – 14
17. Sahin, O., Frohlich, H., Lobke, C., Korf, U., Burmester, S., Majety, M., Mattern, J., Schupp, I., Chaouiya, C., Thieffry, D., Poustka, A., Wiemann, S., Beissbarth, T., Arlt, D.: Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance. BMC Systems Biology **3**(1) (2009)
18. Klamt, S., Saez-Rodriguez, J., Lindquist, J., Simeoni, L., Gilles, E.: A methodology for the structural and functional analysis of signaling and regulatory networks. BMC Bioinformatics **7**(1) (2006) 56
19. Bernot, G., Comet, J.P., Khalis, Z.: Gene regulatory networks with multiplexes. In: European Simulation and Modelling Conference Proceedings. (2008) 423–432