Identification of Biological Regulatory Networks from Process Hitting models

Maxime Folschette, Loïc Paulevé, Katsumi Inoue, Morgan Magnin, Olivier Roux

Identification of Biological Regulatory Networks from Process Hitting models

Maxime Folschette[a,b], Loïc Paulevé[c], Katsumi Inoue[b], Morgan Magnin[a,b], Olivier Roux[a]

[a] *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.*
[b] *National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan.*
[c] *CNRS, Laboratoire de Recherche en Informatique UMR CNRS 8623 Université Paris-Sud, 91405 Orsay Cedex, France*

Abstract

Qualitative formalisms offer a well-established alternative to the more traditionally used differential equation models of Biological Regulatory Networks (BRNs). These formalisms led to numerous theoretical works and practical tools to understand emerging behaviors. The analysis of the dynamics of very large models is however a rather hard problem, which led us to previously introduce the Process Hitting framework (PH), which is a particular class of nondeterministic asynchronous automata network (or safe Petri nets). Its major advantage lies in the efficiency of several static analyses recently designed to assess dynamical properties, making it possible to tackle very large models.

In this paper, we address the formal identification of qualitative models of BRNs from PH models. First, the inference of the Interaction Graph from a PH model summarizes the signed influences between the components that are effective for the dynamics. Second, we provide the inference of all René-Thomas models of BRNs that are compatible with a given PH. As the PH allows the specification of nondeterministic interactions between components, our inference emphasizes the ability of PH to deal with large BRNs with incomplete knowledge on interactions, where Thomas's approach fails because of the combinatorics of parameters.

The inference of corresponding Thomas models is implemented using Answer Set Programming, which allows in particular an efficient enumeration of (possibly numerous) compatible parametrizations.

Keywords: qualitative modeling, model abstraction, interaction graph, René Thomas's parameters, answer set programming

Email address: Maxime.Folschette@irccyn.ec-nantes.fr (Maxime Folschette)

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. Besides continuous models of physicists, generally designed through systems of ordinary differential equations, modeling regulatory networks by means of Boolean networks has become popular in the wake of Kauffman's work [1]. We based our work on a logical formalism developed by René Thomas from 1973 [2] that generalizes upon Boolean networks in the sense that it allows variables to have more than two values and transitions between states to occur asynchronously.

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's 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's 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., [3, 4]), model reduction (e.g., [5]), formal checking of dynamics (e.g., [6, 7]), and the incorporation of time (e.g., [8, 9]) and probability (e.g., [10]) 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, an other major drawback of this framework is the difficulty to specify Thomas's parameters, especially for large networks. Other works have also been carried out with the aim of dealing with large systems (e.g., [11, 12, 13, 14]). Because of the complexity of such systems, these works generally focus on some restricted properties, such as finding steady states.

In order to address the formal analysis of additional dynamical properties (e.g., state reachabilities) within very large BRNs, we recently introduced in [15] a new formalism, named the *Process Hitting* (PH), to model concurrent systems having components with a few qualitative levels. A PH model 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 arity and the effect of synchronizations between components are 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 the qualitative dynamics of BRNs with hundreds or thousands of components, which was impossible with other state-of-the-art approaches [16, 17, 18].

The PH is suitable to model BRNs with different levels of abstraction in the specification of cooperations (associated influences) between components. The concept of cooperation refers to the way two (or more) components jointly influence a third one. In other words, it captures the logical functions stating how various elements coalesce and act together upon an other element among the network. In particular, PH allows to specify partial (nondeterministic) cooperations that superpose the dynamics of candidate fully-specified (deterministic) cooperations. This permits to model BRNs with a partial knowledge on precise evolution functions for components by capturing the largest (the most general) dynamics, without a combinatorics enumeration of compatible Thomas models.

The aim of this paper is to formally establish links between several layers suited for specifying the qualitative dynamics of BRNs, namely: the Interaction Graph (IG), referencing the sign of direct regulations between components; the Process Hitting (PH), which permits to specify the qualitative dynamics of the system with various degrees of knowledge for joint regulations; and Thomas's parameters, which completely specify the functions governing the evolution of components. Another motivation for the work depicted in this paper is that it constitutes an important step to go further in the study of large biological regulatory networks systems beyond simple analysis: it makes it possible to partly control some behaviors. Indeed, when one wants to modify some behaviors (e.g., so as to avoid the reachability of some states or to create stable states) we intend to be able to get upstream the results in accordance at the regulatory level. The outcome of the present work is that we could make such changes on the PH model and then recover afterwards the BRN corresponding to the transformed behavioral system. In a more general framework and in an interesting way, we should be able to synthesize families of BRNs having some featuring behavioral properties.

Firstly, we derive rules that, given a PH model, infer the IG corresponding to the dynamics of the encoded BRN. The obtained IG relates only components that have actual influence in the dynamics. This typically results into IGs that contain less interactions than an hypothetical starting prior IG, as the specification of component evolutions may reveal non-functional regulations. Based on the derived IG, various static analysis allows to conclude on global properties of the system dynamics (see [19] for a short survey). The most known are the Thomas's conjectures (that have been proved since, e.g., [4, 3, 20] for Boolean and discrete networks), which relate the absence of positive cycle to the impossibility for multi-stationarity (distinct attractors), and the absence of negative cycle to the impossibility for oscillatory behaviors.

Second, we tackle the inference of Thomas's parameters that are compatible with a given PH model, i.e., that fully specify the evolution of components while respecting the cooperations specified in PH. More formally, the resulting BRN dynamics is ensured to be included in the PH dynamics: any transition in the

(asynchronous) Thomas model exists in the PH model. In practice, the number of compatible Thomas's parameters can be huge for large networks where only a few cooperations are specified. This highlights that the PH is suitable to analyze dynamics of large-scale networks where the knowledge on cooperations is often incomplete: instead of dealing with a possibly non-tractable number of Thomas models, one can already capture precise dynamical features using a single PH model which gathers all the plausible dynamics.

Finally, we discuss on an implementation of these inference schemes using Answer Set Programming (ASP) [21], which turns out to be effective for these enumerative searches. The whole method is also applied to the model of the bacteriophage lambda immunity control with 4 components, in order to give a detailed application of our method, and to the biological model of the ERBB receptor-regulated G1/S transition, which contains 20 components and allows to show how joint actions can be defined inside a PH model in order to refine its dynamics and study a more precise class of underlying Thomas models.

Our work is related to the approach of [22] which relies on temporal logic, and to [23, 24] that use constraint programming. All three of them 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. 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 [15]) but, on the contrary, in finding out a set of BRNs from one PH.

The contributions presented in this paper significantly extend and improve the preliminary results introduced in [25]. In addition to the improvement of the efficiency and accuracy of the IG inference, we have added support for non-monotonous regulations, that are regulations being positive or negative depending on a particular context. This allows to apply our methodology to a wider class of models. Furthermore, some parts have been completely rewritten in order to explain more precisely the efficiency of the ASP implementation, and give a new detailed biological application of our work.

Outline. Sect. 2 recalls the PH and Thomas frameworks; Sect. 4 defines the IG inference from PH; Sect. 5 details the parameters inference and the enumeration of Thomas's parametrizations compatible with a PH; Sect. 6 discusses the implementation of the latter in ASP. Sect. 7 illustrates our method on a biological model and discusses its applicability on large models.

Notations. $[\![i; j]\!]$ is the set of integers $\{i, i+1, \ldots, j\}$.

2. Frameworks

2.1. The Process Hitting framework

We give 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 $a_i \to b_j \Rsh b_k$ 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 $h = a_i \to b_j \Rsh b_k$ 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 $a \neq b \Rightarrow \forall (a_i, b_j) \in L_a \times L_b, a_i \neq b_j$;
- $\mathcal{H} \triangleq \{a_i \to b_j \Rsh b_k, \dots \mid (a,b) \in \Sigma^2 \wedge (a_i, b_j, b_k) \in L_a \times L_b \times L_b \wedge b_j \neq b_k \wedge a = b \Rightarrow a_i = b_j\}$ 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 $a_i \in s \overset{\Delta}{\Leftrightarrow} s[a] = a_i$. Given a set of sorts $g \subset \Sigma$, $L(g)$ denotes the set of all the sub-states of the sorts in g: $L(g) \triangleq \prod_{b \in g} L_b$.

The set of sorts having an action on a given sort a is noted $\mathsf{hitters}(a)$ (Eq. (1)).

$$\forall a \in \Sigma, \mathsf{hitters}(a) \triangleq \{b \in \Sigma \mid \exists b_i \to a_j \Rsh a_k \in \mathcal{H}\} \tag{1}$$

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 $(s \cdot h \cdot h')$.

PH is a particular sub-class of safe Petri nets [26], i.e., Petri nets with at most one token per place, having groups of mutually exclusive places (processes)

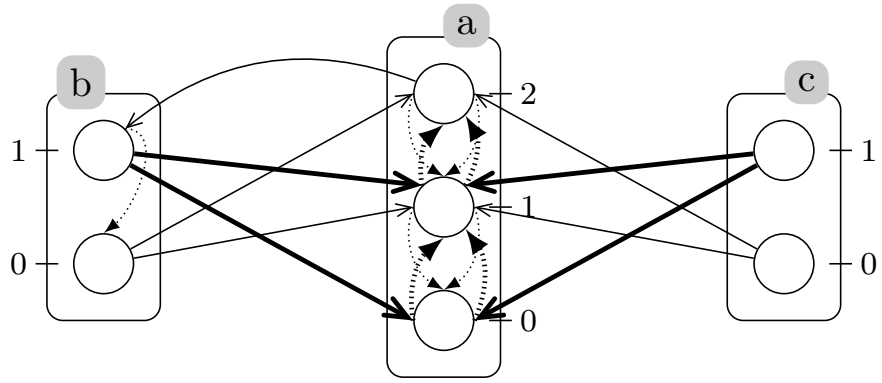


Figure 1: A Process Hitting (PH) example. Sorts are represented by labeled boxes, and processes by circles (ticks are the identifiers of the processes within the sort, for instance, a_0 is the process ticked 0 in the box a). An action (for instance $b_1 \to a_1 \Rsh a_2$) is represented by a pair of 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. Here, actions involving b_1 or c_1 are in thick lines.

acting as the sorts, and where each transition has at most one read-arc (connected to the *hitter*), and exactly one incoming arc (from the *target*) and one outgoing arc (to the *bounce*). The relationship between PH and Petri nets is detailed in [27].

Example 1. 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 (noted a_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 conjunctions. PH restricts the causality of transitions within sorts to the presence of at most one process in another sort (the hitter). One may want to model transitions that should be taken only when several processes are present,

i.e., modeling conjunctions – or cooperations – between the presence of several processes. As described in [15], such cooperations can be encoded in PH with the use of some intermediate sorts, called *cooperative sorts*, that merge information on the state of cooperating sorts. 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 processes of bc 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 sorts b and c hit bc to make it bounce to the process reflecting the state of the sorts b and c (e.g., $b_1 \rightarrow bc_{00} \Rsh bc_{10}$ and $b_1 \rightarrow bc_{01} \Rsh bc_{11}$). Then, it is process bc_{11} which hits a_1 to make it bounce to a_2 instead of 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. It is also worth noticing that the use of such intermediate sorts may trigger spurious transitions due to potential incoherences between the state of the cooperative sort and the actual state of cooperating sorts. However, such phenomena can be fixed by the use of priorities between actions [18].

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 [16] does not suffer from the number of sorts, but on the number of processes within a single sort.

Example 2. 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 c active to reach its highest level, and a does not become inactive unless both b and c are inactive.

2.2. Thomas's modeling

The principle of qualitative modeling was introduced as synchronous Boolean networks by Stuart Kauffman on the one hand [1], and asynchronous René-Thomas networks on the other [2]. Both models have been considered of interest and led to numerous works. In the following, we choose to focus on Thomas's modeling, with its asynchronous, unitary and multi-level semantics.

In this section, we concisely present Thomas's modeling of a Biological Regulatory Network (BRN) and its dynamics, merely inspired by [28, 6, 29].

Thomas's 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 and the conditions of these influences. Then

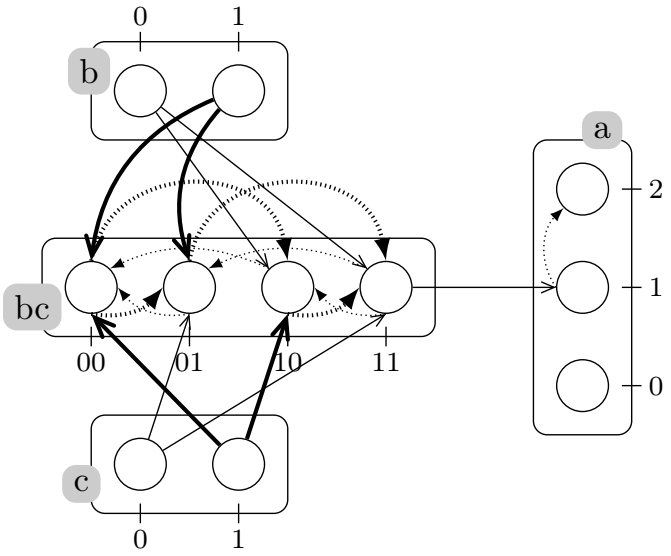


Figure 2: A PH modeling a cooperativity between b_1 and c_1 to make a_1 bounce to a_2. Actions involving b_1 or c_1 are in thick lines.

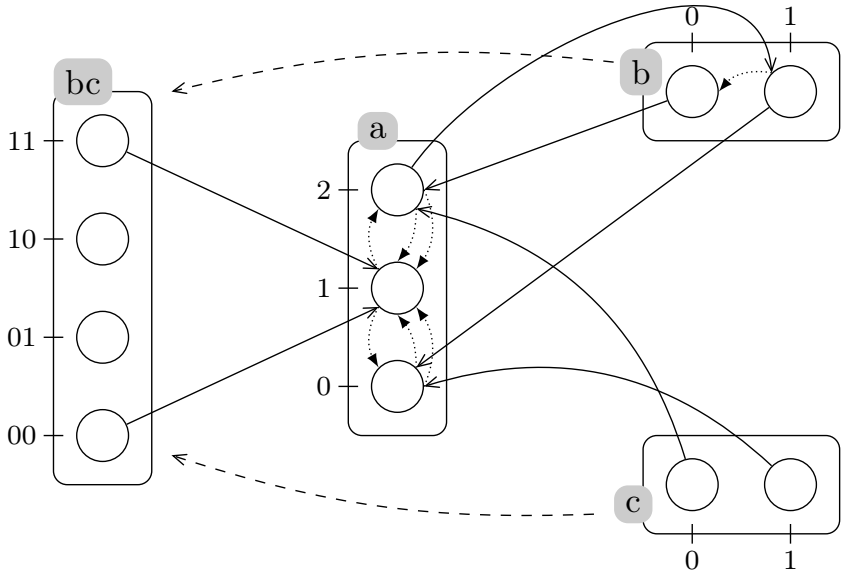


Figure 3: PH resulting from the refinement of the one in Fig. 1 by the specification of several cooperations. The actions from b and c to the cooperative sort bc are identical to those defined in Fig. 2 and are represented here by a single dashed arc.

the *parametrization* removes the ambiguity when a component is targeted by (at least) two different influences. In other words, it specifies the levels towards which a component tends when a given configuration of its regulators applies.

The IG is composed of nodes (a, b, c, ...) that represent components, and edges ($a \xrightarrow{s,t} b$, ...) labeled with a sign (s) and a threshold (t) that stand for regulations between these components (Def. 2). The activity, concentration rate or presence of each component in a given state of the system is modeled by an abstract discrete value called *expression level*. The maximum expression level of a component a is denoted l_a. The sign of an edge denotes the kind of regulation it models: it can be positive ($+$), negative ($-$) or non-monotonous ($\pm$), the latter meaning that the regulation have different sign depending on the context [30]. Regarding the dynamics, an edge $a \xrightarrow{s,t} b$ states that a influences the evolution of b in a certain way when its expression level is above or equal to the threshold t; when its expression level is strictly below this threshold, another kind of influence is expressed, which usually consists of the opposite influence.

The IG is not sufficient to unambiguously define the dynamics of the system. In particular, a component may be regulated by both a positive and negative interaction derived from two different components. To refine the dynamics, a parametrization has been added to the model [31]. This parametrization allows to define focal points, i.e., it clarifies the level towards which a component evolves depending on the expression of its regulators. Some of the information included in the IG and the parametrization may appear as slightly redundant. Especially, if the single parametrization were enriched with additional constraints (e.g., monotonicity requiring an ordering on the value of parameters), then it may directly capture the signs. However, distinguishing the IG and parametrization allows to decouple the graph (which is generally one of the prior knowledge from the biologists) from the constraints on its dynamics. It opened the way to numerous studies (e.g., necessary condition on sustained oscillations) solely based on the structure of the IG (e.g., the analysis of positive or negative circuits).

Definition 2 (Interaction Graph). An *Interaction Graph* (IG) is a couple $\mathcal{G} = (\Gamma, E)$ with Γ the finite set of *components*, and E the finite set of *regulations* between two nodes, labeled with a *sign* and a *threshold*:

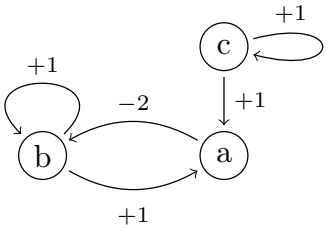
$$E \triangleq \{a \xrightarrow{s,t} b, \dots \mid a, b \in \Gamma \wedge s \in \{+, -, \pm\} \wedge t \in [\![1; l_a]\!]\} \ ,$$

where a regulation from a to b is uniquely referenced:

$$\forall a \xrightarrow{s,t} b \in E, \forall a \xrightarrow{s',t'} b \in E, s = s' \wedge t = t' \ .$$

Given this definition, we denote as a shortcut: $E_s \triangleq \{a \xrightarrow{s,t} b \in E\}$ for $s \in \{+, -, \pm\}$. Furthermore, for all component $b \in \Gamma$, we denote $E^{-1}(b)$ the set of its regulators as defined in Eq. (2).

$$E^{-1}(b) \triangleq \{a \in \Gamma \mid \exists a \xrightarrow{s,t} b \in E\} \tag{2}$$



$$K_{a,\emptyset} = 0 \qquad K_{b,\emptyset} = 0$$
$$K_{a,\{b\}} = 1 \qquad K_{b,\{a\}} = 0$$
$$K_{a,\{c\}} = 1 \qquad K_{b,\{b\}} = 1$$
$$K_{a,\{b,c\}} = 2 \qquad K_{b,\{a,b\}} = 0$$
$$K_{c,\emptyset} = 0 \qquad K_{c,\{c\}} = 1$$

Figure 4: (left) IG example. Components are represented by nodes labeled with a name and regulations by edges labeled with their sign and threshold. For instance, the edge from b to a is labeled $+1$, which stands for: $b \xrightarrow{+1} a$. This means that if the expression level of b is equal to (i.e., above) 1, then b activates a; otherwise, b inhibits a. (right) Example parametrization on the left IG.

Then, for all component a regulating b, i.e., if $a \xrightarrow{s,t} b \in E$, we denote $\mathsf{levels}(a \to b)$ (resp. $\overline{\mathsf{levels}}(a \to b)$) the interval of expression levels of a above (resp. below) the threshold t (Def. 3). For all expression levels of a that belong to $\mathsf{levels}(a \to b)$, a is expected to have the influence corresponding to the sign s on b; for all expression levels belonging to $\overline{\mathsf{levels}}(a \to b)$, the opposite influence is expected. This definition holds because of the uniqueness of the edge $a \xrightarrow{s,t} b$.

Definition 3 (Effective levels (levels)). If $a \xrightarrow{s,t} b \in E$, we define:

$$\mathsf{levels}(a \to b) \triangleq [\![t; l_a]\!] \quad \text{and} \quad \overline{\mathsf{levels}}(a \to b) \triangleq [\![0; t-1]\!] \ .$$

Example 3. Fig. 4(left) represents an Interaction Graph (Γ, E) where $\Gamma = \{a, b, c\}$, with $l_a = 2$ and $l_b = l_c = 1$, and:

$$E_+ = \{b \xrightarrow{+1} a, c \xrightarrow{+1} a, b \xrightarrow{+1} b, c \xrightarrow{+1} c\}$$
$$E_- = \{a \xrightarrow{-2} b\} \qquad E_\pm = \emptyset$$

Hence:

$$E^{-1}(a) = \{b, c\} \qquad E^{-1}(b) = \{a, b\}$$
$$E^{-1}(c) = \{c\}$$

We also have especially:

$$\mathsf{levels}(a \to b) = [\![2; 2]\!] \qquad \overline{\mathsf{levels}}(a \to b) = [\![0; 1]\!]$$

A *state* s of an IG (Γ, E) is an element in $\mathcal{S} \triangleq \prod_{a \in \Gamma} [\![0; l_a]\!]$. $s[a]$ refers to the level of component a in s. For each possible state, the set of *resources* of a given component is the set of regulators of this component whose expression level is

above the threshold of the regulation (Def. 4). In other words, in every state s, a regulator b of a component a is either a resource (if $s[b] \in \mathsf{levels}(b \to a)$) or not a resource (if $s[b] \in \overline{\mathsf{levels}}(b \to a)$). Relying on this observation, the specificity of Thomas's approach lies in the use of discrete *parameters* to represent the focal level interval towards which the component will evolve in every configuration of the resources (Def. 5).

Definition 4 (Resources (Res)). For a given component $a \in \Gamma$ and a state $s \in \mathcal{S}$, the set of regulators of a whose level in s is above the related threshold is called the set of *resources* of a in s and is noted $\mathsf{Res}_a(s)$:

$$\mathsf{Res}_a(s) \triangleq \{b \in E^{-1}(a) \mid s[b] \in \mathsf{levels}(b \to a)\}$$

Definition 5 (Parameter $K_{a,\omega}$ and Parametrization K). For a given component $a \in \Gamma$, and $\omega \subset E^{-1}(a)$ a set of regulators of a, the *parameter* $K_{a,\omega} \in [\![0; l_a]\!]$ is a non-negative integer. The complete map K of parameters on $\mathcal{G}$ is called a *parametrization* on $\mathcal{G}$.

An IG and a parametrization make up a complete BRN. Regarding the dynamics, a parameter $K_{a,\omega}$ is the value towards which a will tend in the states where its resources are exactly the regulators in ω. Indeed, from a given state s, a transition to another state s' is possible provided that exactly one component a evolves of one expression level towards $K_{a,\mathsf{Res}_a(s)}$, as stated by the definition of the transition relation $s \to s'$ (Def. 6). However, a cannot evolve if its expression level already equals the parameter $K_{a,\mathsf{Res}_a(s)}$.

Definition 6 (Asynchronous dynamics ($\to$)). The dynamics of a BRN using Thomas's parameters is given by the transition relation $\to \in \mathcal{S} \times \mathcal{S}$ defined by:

$$\forall s, s' \in \mathcal{S}, s \to s' \iff \exists a \in \Gamma, s[a] \neq K_{a,\mathsf{Res}_a(s)} \wedge s'[a] = s[a] + \delta^a(s)$$
$$\wedge\, \forall b \in \Gamma, b \neq a \Rightarrow s[b] = s'[b]$$

$$\text{with: } \delta^a(s) = \begin{cases} +1 & \text{if } s[a] < K_{a,\mathsf{Res}_a(s)} \\ -1 & \text{if } s[a] > K_{a,\mathsf{Res}_a(s)} \end{cases}$$

Example 4. Fig. 4(right) gives a Parametrization on the IG of Fig. 4(left). In this BRN, the following transitions are possible given the semantics defined in Def. 6:

$$\langle a_0, b_1, c_1\rangle \to \langle a_1, b_1, c_1\rangle \to \langle a_2, b_1, c_1\rangle \to \langle a_2, b_0, c_1\rangle \to \langle a_1, b_0, c_1\rangle \ ,$$

where a_i denotes the component a at level i. This sequence of states ends in a steady state: no evolution is possible in $\langle a_1, b_0, c_1\rangle$.

Remark 1 (Parameters equivalence). In the scope of the asychronous dynamics (Def. 6), one can remark that some parameters values can be equivalent, i.e., the resulting dynamics is the same. It is notably the case with

self-regulations: if $a \xrightarrow{s,t} a$, then for any $\omega \subseteq E^{-1}(a)$, two values v and v' of the parameter $K_{a,\omega\setminus\{a\}}$ are equivalent if $v \geq t$ and $v' \geq t$. Similary, two values v and v' of the parameter $K_{a,\omega\cup\{a\}}$ are equivalent if $v < t$ and $v' < t$.

Interaction Graph of the dynamics. From the dynamics specification of a BRN, one can infer back the IG that contain only the functional regulations. Let us define $f^a(s) = s[a]$ if $s[a] = K_{a,\mathsf{Res}_a(s)}$ and $f^a(s) = s[a] + \delta^a(s)$ otherwise. A positive (resp. negative) interaction from b to a is inferred if there exists a state s such that increasing b in s would increase (resp. decrease) f^a; in other words, if there exists a state s so that: $f^a(s\{b_i\}) <$ (resp. $>$) $f^a(s\{b_{i+1}\})$, where $i < l_b$ and $s\{b_i\}$ denotes the state s where the component b is assigned to i. If all the interactions from b to a are positive (resp. negative), there is a positive (resp. negative) edge from b to a in the IG, denoting a monotonous influence of b on a. In the case when there exists two differently signed interactions, the edge in the IG is said non-monotonous. This abstract representation of the dynamical system can then be used to derive global properties on the dynamics (e.g., [20, 19]).

3. From Local Transitions to Global Functions

Process Hitting (PH) and Thomas modeling rely on two different paradigms for the specification of network dynamics: Thomas's parametrization defines a global function which associates to each component and each resource configuration a level towards which the component eventually evolves in the given configuration. Such a function is always deterministic (whereas the associated asynchronous dynamics can be nondeterministic due to concurrent evolution of components). In contrast, PH models specify a list of local transitions between the levels (processes) of the components (sorts) that are conditioned with the presence of other processes. In addition, PH allows the use of intermediate sorts, such as cooperative sorts, that do not refer to the components of the network.

Hence, identifying Thomas models from PH requires to lump a set of transitions that can be applied in a given configuration into a single process that correspond to the level towards which the component evolves. This step is detailed in Subsect. 3.1 in which is defined the focals function associating the farthest reachable processes, called *focal processes*, of a given sort for a given configuration. It is possible that several different focal processes are identified, indicating an nondeterministic evolution of the sort; and it is also possible that no focal process exist, when in presence of terminal cycles of transitions. In those cases, there is no possible Thomas specification for the PH. The Subsect. 3.2 discusses on constraints for obtaining deterministic (local) behaviors in PH.

Finally, Subsect. 3.3 addresses the separation of PH sorts into components and intermediate cooperative sorts, the latter having to be masked in Thomas models. It notably defines the predecessors, the (groups of) regulators of a sort, and the extension of a configuration to incorporate the state of the cooperative sorts.

3.1. Focal Processes

Given a sort $a \in \Sigma$ and a configuration delimited by a sub-set of processes of sort a and a sub-state σ of the sorts having an action on a, $\mathsf{focals}(a, S_a, \sigma)$ (Def. 7) is the set of processes of sort a towards which a will eventually converge in the scope of the configuration. The configuration delimits the set of states $s \in L$ where $s[a] \in S_a$, and for all sort $b \in \mathsf{hitters}(a), b \neq a$, $s[b] = \sigma[b]$. $\mathsf{focals}(a, S_a, \sigma)$ is obtained from the digraph where $a_i \in S_a$ is connected to $a_j \in L_a$ only if there exists an action $b_k \to a_i \Rsh a_j$ where either $b_k = a_i$ or b_k is in σ. $\mathsf{focals}(a, S_a, \sigma)$ is empty if there is any cyclic terminal connected components; otherwise, it is exactly the leafs of the digraph.

Definition 7 ($\mathsf{focals}(a, S_a, \sigma)$). Given a sort $a \in \Sigma$, a sub-set of its processes $S_a \subset L_a$ and a sub-state $\sigma \in \prod_{b \in \mathsf{hitters}(a), b \neq a} L_b$,

$$\mathsf{focals}(a, S_a, \sigma) \triangleq \begin{cases} \emptyset & \text{if } \mathsf{tscc}(V, E) \neq \emptyset \\ \{a_i \in V \mid \nexists (a_i, a_j) \in E\} & \text{otherwise,} \end{cases}$$

where

$$E \triangleq \{(a_i, a_j) \in (S_a \times L_a) \mid \exists b_k \to a_i \Rsh a_j \in \mathcal{H} : (b_k = a_i \vee \sigma[b] = k)\} \quad (3)$$

$$V \triangleq S_a \cup \{a_j \in L_a \mid \exists (a_i, a_j) \in E\} \quad (4)$$

and $\mathsf{tscc}(V, E)$ are the non-elementary terminal strongly connected components of the digraph (V, E):

$$\mathsf{tscc}(V, E) = \{W \in \mathsf{scc}(V, E) \mid \#W \geq 2 \wedge \forall a_i \in W, (a_i, a_j) \in E \Rightarrow a_j \in W\}$$

with $\mathsf{scc}(V, E)$ the strongly connected components of the digraph (V, E).

From Def. 7 can be derived Property 1 – if $\mathsf{focals}(a, S_a, \sigma)$ is empty, there exists a limit cycle in the evolution of a in the given configuration; and Property 2 – if $\mathsf{focals}(a, S_a, \sigma)$ is non empty, all the evolutions of a in the scope of the given configuration eventually terminate, and the resulting processes are in $\mathsf{focals}(a, S_a, \sigma)$. In other words, if $\mathsf{focals}(a, S_a, \sigma)$ is empty, there exists a sequence of actions that may be played an unbound number of times (cycle) in the given configuration; if it is non-empty, it is ensured that any state in the configuration converges in a bounded number of steps to a process in $\mathsf{focals}(a, S_a, \sigma)$.

Property 1. $\mathsf{focals}(a, S_a, \sigma) = \emptyset$ if and only if there exists a state $s \in L$ where $s[a] \in S_a$ and $\forall b \in \mathsf{hitters}(a), b \neq a, s[b] = \sigma[b]$, such that $\forall n \in \mathbb{N}$ there exists a sequence of actions $h^1, \ldots, h^{n+1}$ in $\mathcal{H}$ sequentially playable in s with $\mathsf{target}(h^1) \in S_a$ and $\forall m \in [\![1; n-1]\!], \mathsf{bounce}(h^m) = \mathsf{target}(h^{m+1})$.

Property 2. If $\mathsf{focals}(a, S_a, \sigma) \neq \emptyset$, for all state $s \in L$ where $s[a] \in S_a$ and $\forall b \in \mathsf{hitters}(a), b \neq a, s[b] = \sigma[b]$, either $\nexists h \in \mathcal{H}$ with $\mathsf{target}(h) \in S_a$ playable in s and $s[a] \in \mathsf{focals}(a, S_a, \sigma)$; or there exists a sequence of actions $h^1, \ldots, h^n$ in $\mathcal{H}$ sequentially playable in s with $\mathsf{target}(h^1) \in S_a$ and $\forall m \in [\![1; n-1]\!], \mathsf{bounce}(h^m) = \mathsf{target}(h^{m+1})$ where $\mathsf{bounce}(h^n) \in \mathsf{focals}(a, S_a, \sigma)$ and where $\nexists h^{n+1} \in \mathcal{H}$ with $\mathsf{target}(h^{n+1}) = \mathsf{bounce}(h^n)$ that is playable in $s \cdot h^1 \cdot \cdots \cdot h^n$.

Example 5. In the PH of Fig. 1, we obtain:

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

3.2. Restrictions for Determinism

As discussed at the beginning of this section, and as formally described in the above sub-section, in the scope of a given configuration, a PH sort may converge to one among several different processes (indeterminism) or may never converge (infinite loop).

We call a *deterministic sort* (Def. 8) a sort a that has a single focal process for each possible configuration of its predecessors in the scope of L_a.

Definition 8 (Deterministic sort). A sort $a \in \Sigma$ is a deterministic sort if and only if each configuration σ of its predecessors leads a to a unique focal process, denoted $a(\sigma)$:

$$\forall \sigma \in L(\mathsf{hitters}(v)), \mathsf{focals}(a, L_a, \sigma) = \{a(\sigma)\}$$

For the inference of the interaction graph and Thomas's parameters from a PH model, the following sections assume that *all the cooperative sorts are deterministic*.

3.3. Separating components from cooperative sorts

The identification of a BRN from a PH assumes that the PH defines two types of sorts: the sorts corresponding to BRN components – noted Γ –, and the cooperative (intermediate) sorts – noted Δ – which should not appear in the BRN. In this subsection, we first give a criteria for identifying the sorts of a PH as either a component or cooperative sort. Then, we characterize the *well-formed* PH for IG inference.

The delimitation 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. This set of sorts in denoted by $\hat{\Gamma}$ (Eq. (5)), whereas the set of cooperative sorts is denoted by $\hat{\Delta}$ (Eq. (6)).

$$\hat{\Gamma} \triangleq \{a \in \Sigma \mid \nexists b_i \to a_j \Rsh a_k \in \mathcal{H}, |j-k| > 1\} \tag{5}$$

$$\hat{\Delta} \triangleq \Sigma \setminus \hat{\Gamma} \tag{6}$$

Given a PH and a partition of its sorts in components Γ and cooperative sorts Δ, Property 3 establishes conditions for BRN identification: in addition of having Γ compatible with $\hat{\Gamma}$ and cooperative sorts being deterministic (Def. 8), 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 3 (Well-formed Process Hitting for BRN identification). A PH is well-formed for BRN identification with $\Sigma = \Gamma \cup \Delta$ only if the following conditions hold:

1. $\Gamma \cap \Delta = \emptyset$, $\Gamma \subseteq \hat{\Gamma}$ and $\forall \upsilon \in \Delta$, υ is a deterministic sort (Def. 8);
2. there is no cycle between cooperative sorts (the digraph $(\Sigma, \{(a,b) \in (\Sigma \times \Sigma) \mid \exists a_i \to b_j \Rsh b_k \in \mathcal{H} \wedge a \neq b \wedge \{a,b\} \cap \Gamma = \emptyset\})$ is acyclic);
3. sorts having no action hitting them belong to Γ ($\{a \in \Sigma \mid \nexists b_i \to a_j \Rsh a_k \in \mathcal{H}\} \subset \Gamma$).

Example 6. In the PH of Fig. 3, bc is a deterministic sort as defined in Def. 8:

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

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

Assuming a PH and a split Γ and Δ of its sort that satisfy Property 3, Def. 9 characterizes the set of predecessors of a sort a as the sorts influencing a through direct actions or intermediate cooperative sorts. The predecessors of a that are components are the regulators of a, denoted $\mathsf{reg}(a)$ (Eq. (9)).

Definition 9 ($\mathsf{pred}(a)$ and $\mathsf{reg}(a)$). Given $a \in \Sigma$, $\mathsf{pred}(a) \subseteq \Sigma$ is the smallest set satisfying the following conditions:

$$\mathsf{hitters}(a) \subseteq \mathsf{pred}(a) \tag{7}$$
$$\upsilon \in \mathsf{pred}(a) \cap \Delta \Rightarrow \mathsf{pred}(\upsilon) \subseteq \mathsf{pred}(a) \tag{8}$$

The components that are predecessors of a are referred to as $\mathsf{reg}(a)$:

$$\mathsf{reg}(a) \triangleq \mathsf{pred}(a) \cap \Gamma \tag{9}$$

As described in the introduction of this section, the identification of Thomas models from PH relies on the inference of focal processes in the different configurations of the regulators of each component. In order to reduce the scope of the configurations that need to be enumerated, we introduce the notion of *group of regulators* of sort a as a set of components that have a joint influence on a. More precisely, b and c are in the same group of regulators of a if there exists an intermediate cooperative sort υ that hits a such that b and c are regulators of υ. Those groups form a partition of $\mathsf{reg}(a)$. We denote $X(a)$ the finest partition in groups of regulators (Def. 10), i.e., where each group is minimal.

Definition 10 (Partition $X(a)$ of $\mathsf{reg}(a)$). Given $a \in \Sigma$, $X(a)$ is the finest partition of $\mathsf{reg}(a)$ such that for any $b, c \in \mathsf{reg}(a)$, $b \neq c$, if there exists a cooperative sort $\upsilon \in \mathsf{hitters}(a) \cap \Delta$ such that $\{b, c\} \subseteq \mathsf{reg}(\upsilon)$ then there exists $g \in X(a)$ such that $\{b, c\} \subseteq g$. If $\mathsf{reg}(a) = \emptyset$, $X(a) \triangleq \{\emptyset\}$.

If the cooperative sorts are all deterministic (Def. 8), it is sufficient to specify the configuration σ of a group g of regulators of a sort a to obtain the configuration of all the sorts hitting a directly. Indeed, because of the absence of cycles between cooperative sorts (Property 3), one can recursively evaluate the focal process of each cooperative sort that hits a with respect to the configuration σ of the regulators g. Such an extended configuration is denoted by $\varsigma_a^g(\sigma)$, formalized in Def. 11

Definition 11 (Configuration extension $\varsigma_a^g(\sigma)$). Given $a \in \Sigma$, a sub-set of regulators $g \subseteq \mathsf{reg}(a)$ and a configuration $\sigma \in L(g)$, $\varsigma_a^g(\sigma)$ is the configuration σ with the corresponding processes of the cooperative sorts in $\mathsf{hitters}(a)$:

$$\varsigma_a^g(\sigma) \triangleq \sigma \uplus \langle \upsilon(\sigma) \mid \upsilon \in \mathsf{hitters}(a) \cap \Delta \wedge \mathsf{reg}(\upsilon) \subseteq g \rangle \tag{10}$$

$$\upsilon(\sigma') \triangleq \begin{cases} \mathsf{focals}(\upsilon, L_\upsilon, \sigma') & \text{if } \mathsf{hitters}(\upsilon) \subseteq \mathsf{dom}(\sigma') \\ \upsilon(\sigma' \uplus \langle \upsilon'(\sigma') \mid \upsilon' \in \mathsf{hitters}(\upsilon) \cap \Delta \rangle) & \text{otherwise,} \end{cases} \tag{11}$$

where $\mathsf{dom}(\sigma')$ is the set of sorts defined in σ' and $\uplus$ denotes the union of two (sub)states.

Example 7. In the PH of Fig. 3, $\varsigma_a^{\{b,c\}}(\langle b_0, c_1 \rangle) = \langle b_0, c_1, bc_{01} \rangle$ and $\varsigma_a^{\{b,c\}}(\langle b_1, c_1 \rangle) = \langle b_0, c_1, bc_{11} \rangle$.

4. Interaction Graph Inference from Process Hitting

The Interaction Graph (IG) is an abstract representation of the direct qualitative influences, positive and/or negative, between the components of the system. As discussed in Sect. 1, the IG allows to efficiently characterize global dynamical properties for the concrete system, such as the capability for multistationarity or oscillation.

The inference of the IG also allows to check for the consistency of the model with respect to prior knowledge on the influences. This is the case for Thomas models, where a prior IG is required to specify the parameters of the dynamics. As detailed in [6], the IG of a dynamical model is consistent with a prior IG if it is a sub-graph of the prior IG. On the one hand, having influences in the model not referenced in the prior IG indicates either mistakes in the encoding of the dynamics, or potential influences that might need to be verified in the biological system. One the other hand, some influences specified in the prior IG may not have been necessary to encode the desired dynamics, indicating some non-minimality of the prior IG.

We consider hereafter a global PH $(\Sigma, L, \mathcal{H})$ and a split of sorts $\Gamma \cup \Delta = \Sigma$ satisfying Property 3 on which the IG inference is to be performed. The inference of the IG is described in Subsect. 4.1, and is illustrated on small examples in Subsect. 4.2.

4.1. Inference of Influences

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, following the standard IG inference from Boolean and discrete networks [20]. This can be seen as looking for changes of derivatives in the dynamics of a when the configuration changes.

Proposition 1 details the inference of all existing influences between components occurring with a threshold t. The inference relies on finding configurations σ of groups g of regulators of a such that the increase of one of the regulators $b \in g$ from b_t to b_{t+1} changes the direction of the evolution of a. If $b \neq a$ (Eq. (12)), this is achieved by finding a configuration σ of g and a such that there exists a bounce a_j from a_i in $\sigma\{b_t\}$ (i.e., the state σ where the process of sort b has been replaced with b_t) which is different from a bounce a_k from a_i in $\sigma\{b_{t+1}\}$. If so, b regulates a at threshold $t+1$ with the sign of $k-j$. The case where $b = a$ (Eq. (13)) is similar: we look for a_t such that there exists a bounce in opposite direction in $\sigma\{a_t\}$ than in $\sigma\{a_{t+1}\}$. If there is no bounce in one of these two configurations, we also need to ensure that the configuration corresponds to a local fixed point between all the regulators of a (instead of in scope of g). The set of configurations where no bounce occur on a_i is given by $\Phi(a_i)$ (Eq. (16)).

Proposition 1 (Influences inference). We define the set of positive (resp. negative) influences $\hat{E}_+$ (resp. $\hat{E}_-$) by

$$\begin{aligned}
&\forall a \in \Gamma, \forall b \in \mathsf{reg}(a), b \neq a, \forall s \in \{+,-\}, \forall t < l_b \\
&\qquad b \xrightarrow{t+1} a \in \hat{E}_s \overset{\Delta}{\Leftrightarrow} \exists g \in X(a), b \in g, \exists \sigma \in L(g \cup \{a\}), \\
&\qquad\qquad \exists a_j \in B_a^g(\sigma\{b_t\}), \exists a_k \in B_a^g(\sigma\{b_{t+1}\}), \\
&\qquad\qquad k > j \wedge s = + \vee k < j \wedge s = -
\end{aligned} \tag{12}$$

$$\begin{aligned}
&\forall a \in \Gamma, \forall s \in \{+,-\}, \forall t < l_a \\
&\qquad a \xrightarrow{t+1} a \in \hat{E}_s \overset{\Delta}{\Leftrightarrow} \exists g \in X(a), \exists \sigma \in L(g \cup \{a\}), \\
&\qquad\qquad \exists a_j \in B_a^{g \cup \{a\}}(\sigma\{a_t\}), \exists a_k \in B_a^{g \cup \{a\}}(\sigma\{a_{t+1}\}), \\
&\qquad\qquad (j = t \Rightarrow \exists \sigma' \in \Phi(a_t) : \sigma\{a_t\} \subseteq \sigma') \\
&\qquad\qquad \wedge (k = t+1 \Rightarrow \exists \sigma' \in \Phi(a_{t+1}) : \sigma\{a_{t+1}\} \subseteq \sigma') \\
&\qquad\qquad \wedge (k \geq t+1 \wedge j \leq t \wedge s = + \\
&\qquad\qquad \vee k < t+1 \wedge j > t \wedge s = -)
\end{aligned} \tag{13}$$

where $B_a^g(\sigma) \subseteq L_a$ is defined as follows:

$$\bar{B}_a^g(\sigma) \overset{\Delta}{=} \{a_j \mid b_k \to a_i \Rsh a_j \in \mathcal{H}, b \in g \wedge b_k \in \varsigma_a^g(\sigma) \wedge a_i \in \sigma\} \tag{14}$$

$$B_a^g(\sigma) \overset{\Delta}{=} \begin{cases} \bar{B}_a^g(\sigma) & \text{if } \bar{B}_a^g(\sigma) \neq \emptyset \\ \{\sigma[a]\} & \text{otherwise} \end{cases} \tag{15}$$

and $\Phi(a_i) \subseteq L(\mathsf{reg}(a) \cup \{a\})$ is defined as follows:

$$\Phi(a_i) \triangleq \{\sigma \in L(\mathsf{reg}(a) \cup \{a\}) \mid a_i \in \sigma \wedge \mathsf{focals}(a, \{a_i\}, \varsigma_a^{\mathsf{reg}(a)}(\sigma)) = \{a_i\}\} \quad (16)$$

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 only exist corresponding influences with the same sign. If an influence is both positive and negative, we infer a non-monotonous edge. In the end, the threshold of each edge is the minimum threshold for which an influence has been found. Note that as regulations match with sign changes in the evolution of the regulated components, if two positive (negative) regulations from b to a have been inferred, necessarily, a third regulation of opposite sign has been inferred in between. The only exception is when a is never hit ($\mathsf{reg}(a) = \emptyset$); in such a case, we arbitrarily pick the minimum threshold (that is, 1) for the self-activation.

Proposition 2 (Interaction Graph inference). We infer $\mathcal{G} = (\Gamma, E)$ using Proposition 1 as follows:

$$\begin{aligned}
E_- &= \{a \xrightarrow{-,t} b \in \hat{E}_- \mid \nexists a \xrightarrow{t'} b \in \hat{E}_+\} \\
E_+ &= \{a \xrightarrow{+,t} b \in \hat{E}_+ \mid \nexists a \xrightarrow{t'} b \in \hat{E}_- \wedge t = \min\{l \mid a \xrightarrow{l} b \in \hat{E}_+\}\} \\
E_\pm &= \{a \xrightarrow{\pm,t} b \mid \exists a \xrightarrow{t'} b \in \hat{E}_+ \wedge \exists a \xrightarrow{t''} b \in \hat{E}_- \\
&\qquad\qquad \wedge t = \min\{l \mid a \xrightarrow{l} b \in \hat{E}_- \cup \hat{E}_+\}\}
\end{aligned}$$

4.2. Examples

The IG inference on the PH of Fig. 3 returns the IG in Fig. 4(left): it finds back all the expected edges alongside with their signs and thresholds. Furthermore, the IG inference on the PH of Fig. 1, which is not refined, gives the same result, despite the absence of the cooperative sort. This is due to the fact that the inference of Proposition 1 is able to analyze the influences from different regulators independently if they do not share a cooperative sort.

If an action $a_2 \to b_0 \not\vdash b_1$ is added to the PH of Fig. 3, then two non-monotonous edges towards b are inferred instead of the previous signed edges:

$$\begin{aligned}
E_+ &= \{b \xrightarrow{+1} a, c \xrightarrow{+1} a, a \xrightarrow{+1} a, c \xrightarrow{+1} c\} \\
E_- &= \emptyset \qquad\qquad E_\pm = \{a \xrightarrow{\pm 2} b, b \xrightarrow{\pm 1} b\}
\end{aligned}$$

This is due to the fact that the actions $a_2 \to b_1 \not\vdash b_0$ and $a_2 \to b_0 \not\vdash b_1$ introduce an oscillation only caused by a, which cannot be represented in Thomas's modeling.

5. Parametrization inference

Given a PH and an IG (either arbitrary, or inferred following the previous section), this section addresses the identification of the parametrization for the corresponding Thomas model.

As described in Subsect. 5.1, this identification relies on the computation of the focal processes for each configuration of the parameters. When there is a unique focal process of a component for a given configuration of its regulators, the focal process matches with the value of the associated Thomas's parameter.

However, in the general case, as described in Sect. 3 and even when the PH is well-formed for parameter inference (characterized in the next subsection by Property 4), there may be several focal processes (nondeterministic behavior) or none (cyclic behavior). Such a case occurs notably when the PH encodes the union of several Boolean or discrete networks, as described in [15, 32].

Hence, we propose in Subsect. 5.2 the notion of *compatible* parametrization with respect ot the PH dynamics: a Thomas model is compatible with a PH if its dynamics is included in the PH dynamics, i.e., all the transitions in the Thomas model are possible transitions in the PH. This relaxed notion of parameterization compatibility allows to enumerate all parametrizations compatible with a given PH, i.e., all Thomas model whose dynamics is included in the PH dynamics.

5.1. Parameters inference

This subsection addresses the inference of independent discrete parameters from a given PH. The inference is equivalent to the one in [15]. In addition, we characterize the well-formed PH for parameter inference property (Property 4), which implies that any process in $\mathsf{levels}(b \to a)$ (resp. $\overline{\mathsf{levels}}(b \to a)$) share the same behavior regarding a.

Property 4 (Well-formed PH for parameter inference). A PH is well-formed for parameter inference if and only if it is well-formed for BRN identification (Property 3) and that the associated IG (Γ, E) verifies that:

$$\begin{aligned}
&\forall b \to a \in E, b \neq a, \forall \sigma \in L(\mathsf{reg}(a)),\\
&\quad \forall (i,j \in \mathsf{levels}(b \to a) \vee i,j \in \overline{\mathsf{levels}}(b \to a)), \forall a_k \in L_a, \qquad (17)\\
&\qquad \mathsf{focals}(a, \{a_k\}, \varsigma_a^{\mathsf{reg}(a)}(\sigma\{b_i\})) = \mathsf{focals}(a, \{a_k\}, \varsigma_a^{\mathsf{reg}(a)}(\sigma\{b_j\}))\\
&\forall a \in \Gamma, \forall b \in \mathsf{reg}(a), b \neq a \wedge b \to a \notin E,\\
&\quad \forall \sigma \in L(\mathsf{reg}(a)), \forall b_i, b_j \in L_b, \forall a_k \in L_a, \qquad (18)\\
&\qquad \mathsf{focals}(a, \{a_k\}, \varsigma_a^{\mathsf{reg}(a)}(\sigma\{b_i\})) = \mathsf{focals}(a, \{a_k\}, \varsigma_a^{\mathsf{reg}(a)}(\sigma\{b_j\}))
\end{aligned}$$

Let $K_{a,\omega}$ be a Thomas's parameter for a given component $a \in \Gamma$ with $\omega \subset E^{-1}(a)$ a set of its regulators. As described in Subsect. 2.2, $K_{a,\omega}$ specifies to which values a eventually evolves in the configuration matching with ω.

The configuration of the PH corresponding to ω is given as follows. For each component $b \in E^{-1}(a)$, we define $\sigma^b_{a,\omega}$ (Eq. (19)) as the process of b with the level in $\mathsf{levels}(b \to a)$ if $b \in \omega$, or in $\overline{\mathsf{levels}}(b \to a)$ if $b \notin \omega$. Because of Property 4,

if several possible levels exist, this process can be chosen arbitrarily. If b does not regulate a in the IG, all processes of sort b should have the same action on a, so the process b_0 is arbitrarily selected (Eq. (20)). The configuration $\sigma_{a,\omega}$ corresponding to ω (Eq. (21)) is then obtained by extending the configuration of the regulators of a to the (deterministic) cooperative sorts (Def. 11 in Subsect. 3.3). Finally, we denote by $S_a(\sigma)$ (Eq. (22)) the set of processes of sort a that are compatible with a configuration σ: if a is specified in σ, then $S_a(\sigma) = \{a_i\}$ where $a_i \in \sigma$; otherwise $S_a(\sigma)$ is the set of all processes of sort a, that is, L_a.

$$\forall b \to a \in E,\ \sigma^b_{a,\omega} \triangleq \begin{cases} \min(\mathsf{levels}(b \to a)) & \text{if } b \in \omega, \\ \min(\overline{\mathsf{levels}}(b \to a)) & \text{if } b \notin \omega \end{cases} \tag{19}$$

$$\forall b \in \mathsf{reg}(a), b \to a \notin E,\ \sigma^b_{a,\omega} \triangleq b_0 \tag{20}$$

$$\sigma_{a,\omega} \triangleq \varsigma_a^{\mathsf{reg}(a)}(\langle \sigma^b_{a,\omega} \mid b \in \mathsf{reg}(a) \rangle) \tag{21}$$

$$\forall a \in \Gamma,\ S_a(\sigma) \triangleq \begin{cases} \{a_i\} & \text{if } a_i \in \sigma \\ L_a & \text{otherwise.} \end{cases} \tag{22}$$

Therefore, we obtain that $K_{a,\omega} = \mathsf{focals}(a, S_a(\sigma_{a,\omega}), \sigma_{a,\omega})$ if this latter is a singleton, denoting a deterministic behavior (Proposition 3).

Proposition 3 (Parameter inference). Let $(\Sigma, L, \mathcal{H})$ be a Process Hitting with $\Sigma = \Gamma \cup \Delta$ and an associated IG $\mathcal{G} = (\Gamma, E)$ well-formed for parameter inference. For all $a \in \Gamma$, for any $\omega \subseteq E^{-1}(a)$, if $\mathsf{focals}(a, S_a(\sigma_{a,\omega}), \sigma_{a,\omega}) = \{a_i\}$, then $K_{a,\omega} = i$.

Example 8. When applied to the refined PH of Fig. 3, and using the IG of Fig. 4(left) which is the result of the IG inference on this PH model (as explained in Subsect. 4.2), Proposition 3 infers the parametrization of Fig. 4(right). The inference of parameters thus finds back all parameters of the original model when the cooperations are fully defined.

Example 9. Regarding the non-refined PH of Fig. 1, and using the same IG of Fig. 4(left), the parameters of Fig. 4(right) are inferred, except the parameters $K_{a,\{b\}}$ and $K_{a,\{c\}}$ for which Proposition 3 is not conclusive. The obtained parametrization is therefore partial. This is due to the lack of precision in the cooperation between b and c, caused by the absence of the cooperative sort.

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 two possible evolutions. Such an indeterminism is not possible in Thomas modeling as in a given configuration of regulators, a component can only have an interval attractor, and can therefore evolve in only one direction. 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 concurrent regulators.

5.2. 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 5) 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 4 holds, any transition in the inferred BRN corresponds to at least one transition in the PH, proving the correctness of our inference. Any parameter that does not satisfy Property 5 is therefore excluded from the enumeration. We remark that all parameters inferred by Proposition 3 satisfy this property.

Property 5 (Parameter validity). A parameter $K_{a,\omega}$ is valid w.r.t. the PH if and only if the following equation is verified:

$$\forall a_i \in C^a_{a,\omega}, a_i \neq K_{a,\omega} \Longrightarrow (\exists c_k \to a_i \not\Rightarrow a_j \in \mathcal{H}, c_k \in C^c_{a,\omega} \\ \wedge a_i < K_{a,\omega} \Rightarrow j > i \wedge a_i > K_{a,\omega} \Rightarrow j < i)$$

Then, we use some additional biological constraints on Thomas's parameters given in [29], that we sum up in the following three properties:

Property 6 (Extreme values assumption). Let $\mathcal{G} = (\Gamma, E)$ be an IG. A parametrization K on $\mathcal{G}$ satisfies the *extreme values assumption* if and only if:

$$\forall b \in \Gamma, E^{-1}(b) \neq \emptyset \Longrightarrow \exists \omega \subset E^{-1}(b), K_{b,\omega} = 0 \wedge \exists \omega' \subset E^{-1}(b), K_{b,\omega'} = l_b$$

Property 7 (Activity assumption). Let $\mathcal{G} = (\Gamma, E)$ be an IG. A parametrization K on $\mathcal{G}$ satisfies the *activity assumption* if and only if:

$$\forall b \in \Gamma, \forall a \in E^{-1}(b), \exists \omega \subset E^{-1}(b), K_{b,\omega} \neq K_{b,\omega \cup \{a\}}$$

Property 8 (Monotonicity assumption). Let $\mathcal{G} = (\Gamma, E)$ be an IG. A parametrization K on $\mathcal{G}$ satisfies the *monotonicity assumption* if and only if:

$$\forall b \in \Gamma, \forall A^+ \subset \{a \in \Gamma \mid a \xrightarrow{+,t} b \in E_+\}, \forall A^- \subset \{a \in \Gamma \mid a \xrightarrow{-,t} b \in E_-\}, \\ K_{b,\omega \cup A^-} \leq K_{b,\omega \cup A^+}$$

Example 10. The parametrization inferred in Example 9 was partial because $K_{a,\{b\}}$ and $K_{a,\{c\}}$ could not be inferred. It is however possible to enumerate all admissible parametrizations compatible with both the inferred parameters,

and the properties of this subsection. This enumeration gives 9 different parametrizations which correspond to the 3 possible values for both $K_{a,\{b\}}$ and $K_{a,\{c\}}$:

$$K_{a,\{b\}} \in \{0,1,2\} \qquad \text{and} \qquad K_{a,\{c\}} \in \{0,1,2\}$$

We note that all these solutions satisfy Properties 5 to 8. All the parametrizations obtained from these combinations are thus admissible.

Finally, we note that the value 1 belongs to the possible values for both parameters. Therefore this enumeration allows, from the model in Fig. 1, to find the behavior of the model refined with a cooperative sort described in Fig. 3.

6. Answer Set Programming implementation concepts

Answer Set Programming (ASP) is a logic programming paradigm [21, 33], which has been chosen to address the enumeration of all admissible parametrizations. ASP can efficiently enumerate a large set of possible answers, and it is easy to constrain the answers according to some properties. Applied to the models and constraints defined in this paper, ASP appears to be efficient for tackling the inherent complexity of the enumeration of parametrizations.

Hereafter, we synthesize some key points of our ASP implementation with the enumeration example.

6.1. Simple rules and answer sets

ASP is based on a set of rules of the form:

$$\underbrace{H}_{head} \leftarrow \underbrace{A_1, A_2, \ldots, A_n, \neg B_1, \neg B_2, \ldots, \neg B_m}_{body}.$$

where the *body* is a series of atoms (A_i) and negations of atoms ($\neg B_i$). In the case of *simple rules* (as opposed to the *cardinality rules* of Subsect. 6.3), the *head* is also an atom (H). Such a rule, whose formal semantics is defined below, schematically states that if all atoms $A_1, A_2, \ldots, A_n$ are true and all atoms $B_1, B_2, \ldots, B_m$ are not true (negation by failure), then H has to be true.

An atom is composed of a predicate and a series of arguments (possibly empty). For example, the following atom:

$$p(x_1, x_2, \ldots, x_r)$$

is composed of the predicate p and r arguments: $x_1, x_2, \ldots, x_r$. Each argument is either a *constant*, which is a representation of a piece of data (component name, expression level, ...), or a *variable*, which is in fact a shorthand for any possible constants (variables are detailed below). In this paper, constants are either numerical or consist of a single lowercase letter (e.g., a, b, c, 1, 2, ...) while variables are always denoted by a single capital letter (e.g., A, P, Q, ...). We do not use function symbols as arguments in this work.

Example 11. Consider a PH model such as depicted in Fig. 1: in order to state the existence of each component, we use a predicate called *component* with two arguments:

$$component(x, n).$$

where x is the name of the component and $l_x = n$ is its maximum expression level.

An ASP program is a set of rules as described above. Solving an ASP program means finding an *answer set*, which is a minimal set of atoms that respect all the rules. In order to formally define this notion of answer set, let us define a *definite rule* as a rule with no negation of atoms (noted "$\neg$" above), and a *definite program* as a program containing only definite rules. Let S be a set of atoms: a definite rule is *satisfied* by S if $H \in S$, or if $\exists i \in [\![1; n]\!], A_i \notin S$. Given this definition of satisfaction, we define the answer set of a definite program Π as the (unique) minimal set S of atoms that satisfies all the rules in Π.

In order to consider the general case, for all non-definite program Π and set S of atoms, we denote Π^S the reduct of Π w.r.t. S, defined from Π by:

1. deleting all the rules that have a negation of atom $\neg B_i$ in the body where $B_i \in S$, and
2. removing all negations of atoms in the bodies of the remaining rules.

Then, a set of atoms S is an answer set of a non-definite program Π if it is the answer set of Π^S, which is a definite program. We note that several answer sets can be solution to the same non-definite program, and in practice a solver can be directed to enumerate them all.

Note that it is possible to define a simple rule with no *body* part. Such a rule is called a *fact*, and its *head* atom consequently has to belong to all answer sets. For instance, the information describing the studied model (the original PH model and the inferred IG and parameters) are expressed in ASP using facts.

Example 12. In order to define the 3 components of Fig. 1, we use the following program:

$$component(a, 2).$$
$$component(b, 1).$$
$$component(c, 1).$$

This program contains only facts using the predicate *component* defined in Example 11, and a, b, c, 1 and 2 are constants.

6.2. Variables

To describe the sets of all expression levels of each component (i.e., the set $[\![0; l_a]\!]$ for each $a \in \Gamma$), one can use atoms of the form $component\_levels(a, k)$ to

state that $k \in [\![0; l_a]\!]$. Variables here come in handy to enumerate each possible constant k for each component a: before solving, any rule containing variables is *grounded*, that is, replaced by an equivalent set of rules with constants only. The following rule, for example, contains three variables (A, K and M) and enumerates the set of possible expression levels of each component in the system:

$$component\_levels(A, K) \leftarrow component(A, M), 0 \leq K \leq M.$$

where the notation "$\leq$" stands for a shortcut in ASP which has the same meaning as the mathematical operator.

Example 13. Regarding Fig. 1, the previous rule together with the facts of Example 12 will give the following answer set:

$$\begin{array}{lll} \{ & component(a,2), & component(b,1), \\ & component(c,1), & component\_levels(a,0), \\ & component\_levels(a,1), & component\_levels(a,2), \\ & component\_levels(b,0), & component\_levels(b,1), \\ & component\_levels(c,0), & component\_levels(c,1). \quad \} \end{array}$$

6.3. Cardinality rules

As an extension of simple rules, *cardinality rules* turn out to be convenient to enumerate a set of answer sets. The head of a cardinality rule specifies a set of atoms H and two integers min and max, and is denoted:

$$min \ \{ \ H \ \} \ max \leftarrow A_1, A_2, \ldots, A_n, \neg B_1, \neg B_2, \ldots, \neg B_m.$$

Given such a rule, as many answer sets as possible are created, so that each answer set S verifies:

$$min \leq |S \cap H| \leq max$$

and every atom $H_i \in S \cap H$ respects the simple rule:

$$H_i \leftarrow A_1, A_2, \ldots, A_n, \neg B_1, \neg B_2, \ldots, \neg B_m.$$

In other words, all answer sets contain a subset of H whose cardinality goes from min to max, and for which the condition in the body of the cardinality rule is met. The set of atoms $H = \{H_1, H_2, \ldots, H_p\}$ is often defined as: $H = \{P \mid Q\}$, which is a shorthand for "the set of atoms of the form P for which Q is true".

Cardinality rules turn out to be convenient to enumerate all possible parametrizations by creating multiple answer sets. For functional purposes, a unique label is assigned to every possible set of resources of a given component. Thus, we denote ω_p the set of resources of a given component a labeled by p, and naturally, K_{a,ω_p} is the related parameter. We note that labeling the sets of resources of a component is obviously equivalent to labeling its parameters. Then, suppose that:

- *param_label*(a, p) states that p is a valid label for a set of resources of component a (and therefore K_{a,ω_p} is a valid parameter);
- *param*(a, p, i) states that: $K_{a,\omega_p} = i$;
- *infered_param*(a, p) states that the parameter inference of K_{a,ω_p} was conclusive (Proposition 3).

It is thereby possible to enumerate the possible values of all parameters for which Proposition 3 was not conclusive, with the following cardinality rule:

$$1 \{ \mathit{param}(A, P, I) \mid \mathit{component\_levels}(A, I) \} 1 \leftarrow \mathit{param\_label}(A, P), \neg \mathit{infered\_param}(A, P).$$

Indeed, this rule applies to any possible parameter P of any component A (*param_label*) whose value is still unknown ($\neg$*infered_param*), and states that any expression level I of this component (*component_levels*) is a candidate value for the parameter (*param*). Furthermore, the lower and upper bounds are both 1, which forces each enumerated parameter to have exactly one value. In other worlds, this cardinality rule creates as many answer sets as there are *candidate* parametrizations so that if K_{a,ω_p} could not be inferred by Proposition 3, then $K_{a,\omega_p} \in [\![0; l_a]\!]$ (thus completely disregarding the notion of admissible parametrizations given in Subsect. 5.2).

Example 14. In the scope of Example 9, $K_{a,\{b\}}$ and $K_{a,\{c\}}$ could not be inferred by Proposition 3. The previous cardinality rule allows to produce 9 parametrizations, in which these two parameters can take all possible values:

$$(K_{a,\{b\}}; K_{a,\{c\}}) \in \{0, 1, 2\} \times \{0, 1, 2\}$$

and all the other parameters keep their inferred values.

6.4. Constraints

Finally, a *constraint* is a rule with no *head* part:

$$\leftarrow A_1, A_2, \ldots, A_n, \neg B_1, \neg B_2, \ldots, \neg B_m.$$

A constraint is satisfied only if its *body* is not satisfied, which thus allows to invalidate answer sets containing some unwanted combinations of atoms. In the scope of parameters enumeration, for example, constraints are especially useful to filter out parametrizations that do not respect the assumptions of Subsect. 5.2. Indeed, suppose that:

- *less_active*(a, p, q) states that ω_p is a set of resources of a with (loosely) less activators and more inhibitors than ω_q;
- *param_inf*(a, p, q) states that: $K_{a,\omega_p} \leq K_{a,\omega_q}$.

Then, the monotonicity assumption (Property 8) is formulated as the following constraint:

$$\leftarrow less\_active(A, P, Q), \neg param\_inf(A, P, Q).$$

Indeed, this constraint removes all parametrization results where parameters K_{A,ω_P} and K_{A,ω_Q} exist such that A is less activated by the set of resources ω_P than it is by ω_Q, but $K_{A,\omega_Q} < K_{A,\omega_P}$, thus violating the monotonicity assumption. Of course, other assumptions can be formulated in the same way.

Example 15. All the candidate values enumerated in Example 14 already respect Properties 5 to 8. Therefore, the constraints encoding these properties do not filter any solution.

This subsection succinctly described how ASP programs come in handy to represent a model and solve complex problems on it. 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 candidate parametrizations to be considered in the end. However, all steps of the inference presented in this paper (Sect. 4 & 5) were implemented in and benefited from this programming paradigm.

7. Examples

This section aims at giving several applications of our work in order to understand its results and range of applications. First, Subsect. 7.1 gives a detailed application of our IG inference method by studying parts of a model of the phage lambda immunity control. Then, Subsect. 7.2 gives a practical application of our results on a biological model of epithelial growth factor receptor taken from the literature. Finally, Subsect. 7.3 gives data related to the results and computation times of our software when applied to several large biological models.

The inference and enumeration methods described in this paper have been implemented as part of PINT[1], which gathers PH related tools. Our implementation mainly consists in ASP programs that are solved using Clingo[2]. The IG and parameter inferences can be performed using the following command:

```
ph2thomas -i model.ph --dot ig.dot
```

where `model.ph` is the PH model file in PINT format, and `ig.dot` is an output file to write the inferred IG in DOT format. The (possibly partial) inferred parametrization will be returned on the standard output. Instead of the `--dot`

[1] Available at `http://loicpauleve.name/pint`
[2] Available at `http://potassco.sourceforge.net`

`ig.dot` option, it is possible to specify an alternative IG to perform the parameters inference on (instead of using the inferred IG) with the `--ig model.ig` option, where `model.ig` is an IG model file in PINT format. The admissible parametrizations enumeration is performed by appending the `--enumerate` option to the command.

7.1. Detailed example: the bacteriophage lambda immunity control

In order to illustrate the IG inference developed as Sect. 4, this first subsection focuses on a model of the lambda phage immunity control, which has been widely studied mainly for its particular switch mechanism, allowing this virus to "chose" between lysis and lysogenization. These two possible responses are characterized by different dynamics that lead to two separate attractors.

We consider here the model with four genes proposed in [34], which will serve as example to detail the inference method proposed in the previous sections. This model can be straightforwardly represented under the form of a Process Hitting with 4 components and 4 cooperative sorts:

$$\Sigma = \{\underbrace{\mathsf{cI}, \mathsf{cro}, \mathsf{cII}, \mathsf{N}}_{\Gamma}, \underbrace{\mathsf{cI\text{-}cro\text{-}cII}, \mathsf{cI\text{-}cro\text{-}N}, \mathsf{cI\text{-}cro}, \mathsf{cro\text{-}cII}}_{\Delta}\}$$

The four components have respectively 3, 4, 2 and 2 expression levels, which we denote:

$$L_{\mathsf{cI}} = \{\mathsf{cI}_0, \mathsf{cI}_1, \mathsf{cI}_2\} \qquad L_{\mathsf{cro}} = \{\mathsf{cro}_0, \mathsf{cro}_1, \mathsf{cro}_2, \mathsf{cro}_3\}$$
$$L_{\mathsf{cII}} = \{\mathsf{cII}_0, \mathsf{cII}_1\} \qquad L_{\mathsf{N}} = \{\mathsf{N}_0, \mathsf{N}_1\}$$

The full Process Hitting model is not represented here due to a lack of space, but excerpts are given in Fig. 5 and Fig. 6. The whole set of actions is given in appendix.

7.1.1. Regulation of cI *on* N

We first study the influence of cI on N. The relevant part of the Process Hitting model for this problem is depicted in Fig. 5. It involves a third component cro and the cooperative sort cI-cro; indeed, by studying the full model, we find:

$$\mathsf{pred}(\mathsf{N}) = \{\mathsf{cI}, \mathsf{cro}, \mathsf{cI\text{-}cro}\} \qquad \mathsf{reg}(\mathsf{N}) = \{\mathsf{cI}, \mathsf{cro}\} \qquad X(\mathsf{N}) = \{\{\mathsf{cI}, \mathsf{cro}\}\}$$

Thus, we only have to focus on the group of regulators $\{\mathsf{cI}, \mathsf{cro}\}$ in order to analyze the regulation $\mathsf{cI} \to \mathsf{N}$. The idea behind the IG inference developed in Sect. 4 is to make the level of the considered regulator vary (here, cI) while keeping constant all the other regulators in the same group (here, only cro) and to observe the results on the evolution of the observed component (here, N). In other words, if increasing the sole level of cI makes N tend to decrease, then a negative local influence of cI on N is inferred.

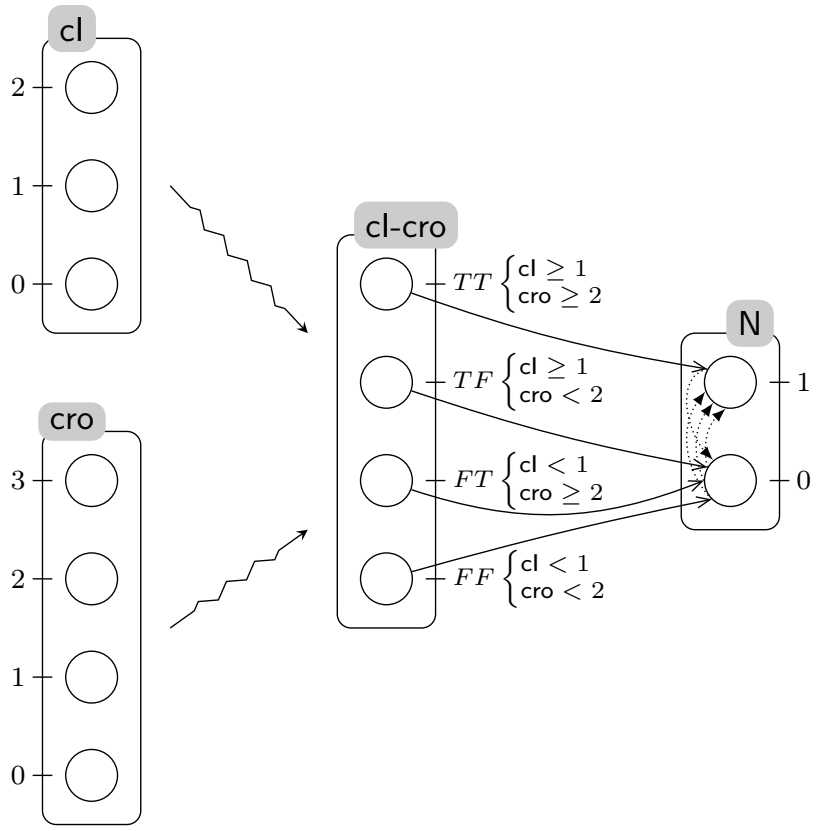


Figure 5: A portion of the PH model used in this section modeling the phage lambda immunity response. This part of the whole model only depicts the interactions towards N from components cI and cro. All the actions updating the cooperative cI-cro are simply depicted by two zigzag arrows. Furthermore, as this cooperative sort cI-cro is deterministic, its processes are labeled by the configuration they represent, given by two Boolean variables indicating whether the values of cI and cro lie above their respective thresholds (T) or not (F). For example, the configuration $\langle \mathsf{cI}_1, \mathsf{cro}_0 \rangle$ is represented by the process TF.

We first note that in the case where cro_3 is present, we have from Eq. (10):

$$\varsigma_{\mathsf{N}}^{\{\mathsf{cl},\mathsf{cro}\}}(\langle \mathsf{cl}_0, \mathsf{cro}_3, \mathsf{N}_0\rangle) = \langle \mathsf{cl}_0, \mathsf{cro}_3, \mathsf{N}_0, \mathsf{cl\text{-}cro}_{FT}\rangle$$
$$\varsigma_{\mathsf{N}}^{\{\mathsf{cl},\mathsf{cro}\}}(\langle \mathsf{cl}_1, \mathsf{cro}_3, \mathsf{N}_0\rangle) = \langle \mathsf{cl}_1, \mathsf{cro}_3, \mathsf{N}_0, \mathsf{cl\text{-}cro}_{TT}\rangle$$
$$\varsigma_{\mathsf{N}}^{\{\mathsf{cl},\mathsf{cro}\}}(\langle \mathsf{cl}_2, \mathsf{cro}_3, \mathsf{N}_0\rangle) = \langle \mathsf{cl}_2, \mathsf{cro}_3, \mathsf{N}_0, \mathsf{cl\text{-}cro}_{TT}\rangle$$

We can then deduce the following directions of evolution from Eq. (15):

$$B_{\mathsf{N}}^{\{\mathsf{cl},\mathsf{cro}\}}(\langle \mathsf{cl}_0, \mathsf{cro}_3, \mathsf{N}_0\rangle) = \{\mathsf{N}_1\}$$
$$B_{\mathsf{N}}^{\{\mathsf{cl},\mathsf{cro}\}}(\langle \mathsf{cl}_1, \mathsf{cro}_3, \mathsf{N}_0\rangle) = \{\mathsf{N}_0\}$$
$$B_{\mathsf{N}}^{\{\mathsf{cl},\mathsf{cro}\}}(\langle \mathsf{cl}_2, \mathsf{cro}_3, \mathsf{N}_0\rangle) = \{\mathsf{N}_0\}$$

Therefore, Eq. (12) gives: $\mathsf{cl} \xrightarrow{1} \mathsf{N} \in \hat{E}_-$. The configurations using the other processes of cro (cro_0, cro_1 and cro_2) do not add more information on the influences towards N. Thus, given Proposition 2, it comes: $\mathsf{cl} \xrightarrow{-1} \mathsf{N} \in E$.

We note that an equivalent work on cro also gives the second influence on N: $\mathsf{cro} \xrightarrow{-2} \mathsf{N} \in E$.

7.1.2. Self-regulation of cro *due to* cl

We now wish to study the self-regulation of cro. For this, we will show that studying only the actions self-hitting cro is not enough; indeed, it is required to also study the interactions with cl. We have the following results from the whole model:

$$\mathsf{pred}(\mathsf{cro}) = \{\mathsf{cl}, \mathsf{cro}\} \qquad \mathsf{reg}(\mathsf{cro}) = \{\mathsf{cl}, \mathsf{cro}\} \qquad X(\mathsf{cro}) = \{\{\mathsf{cl}\}, \{\mathsf{cro}\}\}$$

The interactions of cro and cl on cro have been represented in Fig. 6.

Let us first focus on the group of regulators $\{\mathsf{cro}\}$. Of course:

$$\forall \mathsf{cro}_i \in L_{\mathsf{cro}}, \varsigma_{\mathsf{cro}}^{\{\mathsf{cro}\}}(\langle \mathsf{cro}_i\rangle) = \langle \mathsf{cro}_i\rangle$$

and from Eq. (15) and (16):

$$B_{\mathsf{cro}}^{\{\mathsf{cro}\}}(\langle \mathsf{cro}_3\rangle) = \{\mathsf{cro}_2\}$$
$$B_{\mathsf{cro}}^{\{\mathsf{cro}\}}(\langle \mathsf{cro}_2\rangle) = \{\mathsf{cro}_1\} \quad \text{and} \quad \Phi(\mathsf{cro}_2) = \emptyset$$
$$B_{\mathsf{cro}}^{\{\mathsf{cro}\}}(\langle \mathsf{cro}_1\rangle) = \{\mathsf{cro}_1\} \quad \text{and} \quad \Phi(\mathsf{cro}_1) = \emptyset$$
$$B_{\mathsf{cro}}^{\{\mathsf{cro}\}}(\langle \mathsf{cro}_0\rangle) = \{\mathsf{cro}_0\} \quad \text{and} \quad \langle \mathsf{cl}_2, \mathsf{cro}_0\rangle \in \Phi(\mathsf{cro}_0)$$

We thus cannot infer any local influence from Eq. (13) by observing the group of regulators $\{\mathsf{cro}\}$, because $\Phi(\mathsf{cro}_1) = \Phi(\mathsf{cro}_2) = \emptyset$. This is due to the fact that all processes of cl can hit cro_1 and cro_2; these processes can therefore not be considered stable in any configuration.

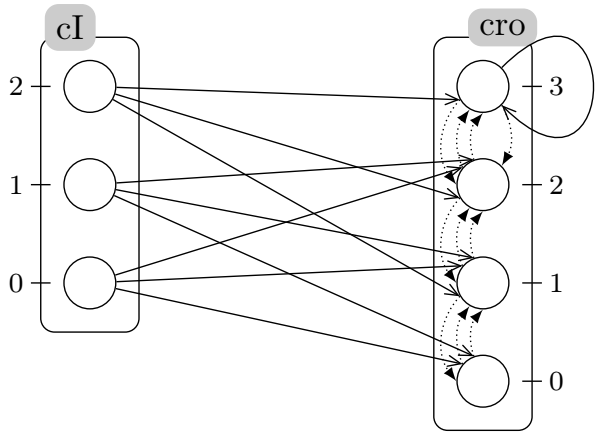


Figure 6: A portion of the PH model used in this section modeling the phage lambda immunity response. This part of the whole model only depicts the interactions towards cro from components cI and cro.

Let us now focus instead on the group of regulators $\{\mathsf{cI}\}$. One again, we trivially have:

$$\forall \mathsf{cI}_i \in L_{\mathsf{cI}}, \forall \mathsf{cro}_j \in L_{\mathsf{cro}}, \varsigma_{\mathsf{cro}}^{\{\mathsf{cI}\}}(\langle \mathsf{cI}_i, \mathsf{cro}_j \rangle) = \langle \mathsf{cI}_i, \mathsf{cro}_j \rangle$$

Furthermore, from Eq. (15), we especially find:

$$B_{\mathsf{cro}}^{\{\mathsf{cI}\}}(\langle \mathsf{cI}_0, \mathsf{cro}_2 \rangle) = \{\mathsf{cro}_3\}$$
$$B_{\mathsf{cro}}^{\{\mathsf{cI}\}}(\langle \mathsf{cI}_0, \mathsf{cro}_3 \rangle) = \{\mathsf{cro}_2\}$$

Therefore, Eq. (12) gives: $\mathsf{cro} \xrightarrow{3} \mathsf{cro} \in \hat{E}_-$. Finally, as no more information is added by observing the other levels of cro and cI, and given Proposition 2, it comes: $\mathsf{cro} \xrightarrow{-3} \mathsf{cro} \in E$.

7.1.3. Conclusion on the phage lambda immunity control

The IG inference of Proposition 2 applied to the whole Process Hitting modeling the phage lambda immunity response produces the IG given in Fig. 7. This IG contains all edges included in the original model except the self-influence on cI (namely, $\mathsf{cI} \xrightarrow{+2} \mathsf{cI} \notin E$); this is due to the fact that an equivalent dynamics exists without this edge.

Finally, we note that the parameters inference of Proposition 3 applied to the same model, and using the output of the previous IG inference is conclusive for all parameters. The complete parametrization produced is available in Table 1.

We note two main differences between the parametrization obtained by our inference and the one proposed in [34]:

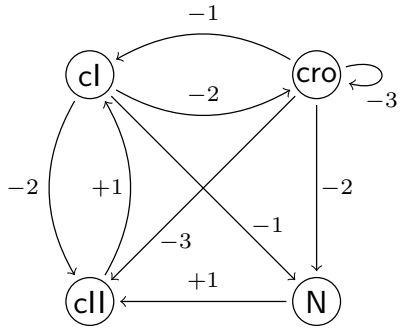


Figure 7: Result of the IG inference performed on the Process Hitting model of the phage lambda immunity response taken from [34].

- The absence of the self-regulation of cI automatically reduces the size of the parametrization, but this does not modify the dynamics, as mentioned previously;
- Our inference returns $K_{\mathsf{cro},\{\mathsf{cI};\mathsf{cro}\}} = 2$, which is a value equivalent to the original value 0 according to Remark 1: the resulting dynamics is the same, and 1 would also have been an acceptable value.

7.2. Biological application: the epithelial growth factor receptor

This subsection focuses on the study of the epidermal growth factor (EGF) receptor model detailed in [35]. This model is represented by an IG containing 20 components and 52 edges. A protein named EGF, having no regulator, can be considered as the only input, and a chain of reactions leads to the activation of protein pRB which is responsible for regulating the cell division, therefore making it essential to prevent cancer development.

Three models are created from the original IG, with different levels of precision regarding the cooperation between components. The translation from an IG into a Process Hitting is not detailed here as it was previously covered in [15]. Model (1) represents a translation of the raw IG into Process Hitting, that is, without any knowledge of the Boolean rules (and therefore the cooperations) of the components. Model (2) implements some of the rules based on the results of several knockdown experiments. Model (3) is the totally refined model with all cooperations implemented given the Boolean functions of all components. Therefore, those three models can be considered as successive refinements of the original and most general one. The results of the IG and parameters inferences are detailed in Table 2 and discussed in the following alongside with details about their construction.

Model (1) encompasses only sole interactions between components, that is, independent activations or inhibitions of a component on another given the regulations specified in the original IG. Therefore, the IG inferred from model (1) is the same as the IG used to create the model, with one additional positive

$$
\begin{aligned}
K_{\mathsf{cI},\emptyset} &= 2 & K_{\mathsf{N},\emptyset} &= 1\\
K_{\mathsf{cI},\{\mathsf{cro}\}} &= 0 & K_{\mathsf{N},\{\mathsf{cro}\}} &= 0\\
K_{\mathsf{cI},\{\mathsf{cII}\}} &= 2 & K_{\mathsf{N},\{\mathsf{cI}\}} &= 0\\
K_{\mathsf{cI},\{\mathsf{cII};\mathsf{cro}\}} &= 2 & K_{\mathsf{N},\{\mathsf{cI};\mathsf{cro}\}} &= 0
\end{aligned}
$$

$$
\begin{aligned}
K_{\mathsf{cII},\emptyset} &= 0 & K_{\mathsf{cro},\emptyset} &= 3\\
K_{\mathsf{cII},\{\mathsf{cro}\}} &= 0 & K_{\mathsf{cro},\{\mathsf{cro}\}} &= 2\\
K_{\mathsf{cII},\{\mathsf{cI}\}} &= 0 & K_{\mathsf{cro},\{\mathsf{cI}\}} &= 0\\
K_{\mathsf{cII},\{\mathsf{cI};\mathsf{cro}\}} &= 0 & K_{\mathsf{cro},\{\mathsf{cI};\mathsf{cro}\}} &= 2\\
K_{\mathsf{cII},\{\mathsf{N}\}} &= 1\\
K_{\mathsf{cII},\{\mathsf{N};\mathsf{cro}\}} &= 0\\
K_{\mathsf{cII},\{\mathsf{N};cI\}} &= 0\\
K_{\mathsf{cII},\{\mathsf{N};cI;\mathsf{cro}\}} &= 0
\end{aligned}
$$

Table 1: Result of the parameters inference performed on the Process Hitting model of the phage lambda immunity response taken from [34].

auto-edge on the only input EGF (which is due to its absence of regulators). The only parameters that could be inferred are the parameters for the extreme cases of regulation (all activators present and all inhibitors absent, and the opposite). This first model therefore abstracts a large number of Thomas models as a lot of parameters are left undecided.

In order to build model (2), 14 cooperative sorts were added in order to model the Boolean functions of several components (consisting of AND and OR operands). To do so, the following components were noticed due to their importance in the chain of reactions: CDK4, CDK6, CycD1, ERα and c-MYC. Indeed, knockdown experiments have been conducted in [35] and the results showed that knocking down these components lead to an important decrease in the production of pRB. We therefore concluded that these components were involved in other components' Boolean functions in a way that the knockdown of the former was sufficient to prevent the production of the latter (which is typical of AND operands). In order to reproduce such requirements, the Boolean functions of their successors, that are CDK4, CDK6, prB, p21, p27, IGF1R, MYC, CycD1 and CycE1, were modeled as cooperative sorts, if needed. In theory, 9 cooperative sorts would have sufficed, but the chaining of cooperative sorts described in Subsect. 2.1 was used to reduce the number of processes in each cooperative sort. As a result, the added cooperations allowed to infer about half the parameters; however, the number of possible Thomas models that can be inferred from this PH is still significant because of the numerous remaining unknown parameters. Furthermore, we note that the inferred IG contains one edge less than the original IG. This is due to the fact that one of the Boolean

Model	|E|	|K|	Inferred parameters	Possible models	Fixed points
(1)	52	196	20	$2^{176} \simeq 9.6 \cdot 10^{52}$	0
(2)	51	192	98	$2^{94} \simeq 2.0 \cdot 10^{28}$	0
(3)	51	192	192	1	3

Table 2: Results of the IG and parameters inference on three models derived from the EGF receptor model of [35] with different precisions in the definition of the cooperations. Model (1) contains no cooperations between the components. Several cooperations were included in model (2) under the form of 14 cooperative sorts and all of them were included in Model (3) under the form of 22 cooperative sorts. The second column gives the number of edges in the IG inferred with Proposition 2 (the number of nodes is always the number of components in the model, that is, 20). The third column gives the number of parameters in the model (given the IG), the fourth column gives the number of parameters that could be inferred using Proposition 3, and the fifth column consequently gives the number of compatible models with the studied PH model, which exponentially depends on the number of parameters that could not be inferred. Finally, the last column gives the number of fixed points in the PH model, computed with another existing PH tool provided with Pint.

functions could in fact be simplified in a way that a component did not appear anymore in it. No edge have therefore been inferred by our method in this case.

Finally, model (3) was build using all the Boolean functions provided in [35]. These functions take the form of 22 cooperative sorts into the model in order to match the desired behavior of the system. As all cooperations are fully defined in this model, all the parameters are inferred and only one Thomas model can be derived. We note also that this PH model is the only one containing at least one fixed point. In fact, the three found steady states include the two states that correspond to a complete propagation of the input signal, that is, in the case where EGF is active and in the case where it is not. The two other models contain no fixed point because some cooperations are not fully defined, leading to oscillations that are a consequence of the nondeterministic behavior.

7.3. Computation times on several large models

The current implementation can successfully handle large PH models of BRNs found in the literature such as:

- the EGF receptor model from [35] with 20 components presented in the last subsection[3],
- a T cell receptor model described as an IG in [36], which contains 40 components and 14 cooperative sorts.

For each model, IG and parameters inferences are performed together in less than a second on a standard desktop computer.

Bigger models related to the aforementioned systems were also tested with our implementation:

[3] All models mentioned in this section are available as examples distributed with Pint.

- a model of the T cell receptor with 94 components, described in [37],
- a model of the EGF receptor with 104 components, described in [38].

These two models were obtained in a previous work by an automatic translation from the CellNetAnalyzer [39] formalism.

The composition of all models and the results of the inferences are summarized in Table 3. In every case, all parameters could be inferred and it was not necessary to enumerate compatible parametrizations.

Model	$\lvert\boldsymbol{\Gamma}\rvert$	$\lvert\boldsymbol{\Delta}\rvert$	**IG inference**	**K inference**	$\lvert\mathbf{K}\rvert$
EGF receptor [35]	20	22	<1s	<1s	192
T cell receptor [36]	40	14	<1s	<1s	143
T cell receptor [37]	94	39	100s	<1s	578
EGF receptor [38]	104	89	200s	2.5s	27496

Table 3: Computation times and several pieces of information related to the IG and parametrization inferences of four biological models. The second column gives the number of components of each model and the third column gives the number of cooperative sorts used to model joint actions. The fourth (resp. fifth) column gives the computation times of the IG inference (resp. the parametrization inference). The last column gives the number of parameters in each model.

8. Conclusion and Discussion

This work establishes the abstraction relationship between PH and Thomas's approaches for qualitative BRN modeling. The PH allows an abstract representation of BRNs dynamics (allowing incomplete knowledge on the cooperation between components) that cannot be exactly represented in René Thomas's 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's parametrization for BRNs from a PH model, we delimit the set of admissible Thomas's 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 with several results on both small and large biological models.

Several extensions of the presented work are now to be considered. First, the link between successively refined models of a system could be formally studied.

Indeed, it is convenient to refine a PH model by removing actions or adding cooperations; the study and formalization of such process would allow to predict behavioral changes and lead to more accurate models. Second, the inference of BRN multiplexes [40] 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 cannot 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.

Acknowledgement. This work was partially supported by the Fondation Centrale Initiatives and the French National Agency for Research (ANR-10-BLANC-0218 BioTempo project).

[1] S. A. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets, Journal of theoretical biology 22 (3) (1969) 437–467.

[2] R. Thomas, Boolean formalization of genetic control circuits, Journal of Theoretical Biology 42 (3) (1973) 563 – 585.

[3] A. Richard, J.-P. Comet, Necessary conditions for multistationarity in discrete dynamical systems, Discrete Applied Mathematics 155 (18) (2007) 2403 – 2413.

[4] É. Remy, P. Ruet, D. Thieffry, Graphic requirements for multistability and attractive cycles in a boolean dynamical framework, Advances in Applied Mathematics 41 (3) (2008) 335 – 350.

[5] A. Naldi, E. Remy, D. Thieffry, C. Chaouiya, A reduction of logical regulatory graphs preserving essential dynamical properties, in: Computational Methods in Systems Biology, Vol. 5688 of LNCS, Springer, 2009, pp. 266–280.

[6] A. Richard, J.-P. Comet, G. Bernot, Modern Formal Methods and Applications, 2006, Ch. Formal Methods for Modeling Biological Regulatory Networks, pp. 83–122.

[7] A. Naldi, D. Thieffry, C. Chaouiya, Decision diagrams for the representation and analysis of logical models of genetic networks, in: Computational Methods in Systems Biology, Springer, 2007, pp. 233–247.

[8] H. Siebert, A. Bockmayr, Incorporating time delays into the logical analysis of gene regulatory networks, in: Computational Methods in Systems Biology, Vol. 4210 of LNCS, Springer, 2006, pp. 169–183.

[9] J. Ahmad, O. Roux, G. Bernot, J.-P. Comet, A. Richard, Analysing formal models of genetic regulatory networks with delays, International Journal of Bioinformatics Research and Applications (IJBRA) 4 (3) (2008) 240–262.

[10] S. Twardziok, H. Siebert, A. Heyl, Stochasticity in reactions: a probabilistic boolean modeling approach, in: Computational Methods in Systems Biology, ACM, 2010, pp. 76–85.

[11] F. Corblin, S. Tripodi, E. Fanchon, D. Ropers, L. Trilling, A declarative constraint-based method for analyzing discrete genetic regulatory networks, Biosystems 98 (2) (2009) 91 – 104.

[12] D. Bollman, O. Colon-Reyes, E. Orozco, Fixed points in discrete models for regulatory genetic networks, EURASIP Journal on Bioinformatics and Systems Biology 2007 (1) (2007) 97356.

[13] F. Delaplace, H. Klaudel, T. Melliti, S. Sené, Analysis of modular organisation of interaction networks based on asymptotic dynamics, in: D. Gilbert, M. Heiner (Eds.), Computational Methods in Systems Biology, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 148–165.

[14] F. Ay, F. Xu, T. Kahveci, Scalable steady state analysis of boolean biological regulatory networks, PLoS ONE 4 (12) (2009) e7992.

[15] L. Paulevé, M. Magnin, O. Roux, Refining dynamics of gene regulatory networks in a stochastic π-calculus framework, in: Transactions on Computational Systems Biology XIII, Springer, 2011, pp. 171–191.

[16] L. Paulevé, M. Magnin, O. Roux, Static analysis of biological regulatory networks dynamics using abstract interpretation, Mathematical Structures in Computer Science 22 (04) (2012) 651–685.

[17] L. Paulevé, G. Andrieux, H. Koeppl, Under-approximating Cut Sets for Reachability in Large Scale Automata Networks, in: CAV'13 - Computed Aided Verification, 2013, accepted.

[18] M. Folschette, L. Paulevé, M. Magnin, O. Roux, Under-approximation of reachability in multivalued asynchronous networks, in: Proceedings of the 4th International Workshop on Interactions between Computer Science and Biology (CS2Bio'13), Electronic Notes in Theoretical Computer Science, 2013, accepted.

[19] L. Paulevé, A. Richard, Static analysis of boolean networks based on interaction graphs: a survey, Electronic Notes in Theoretical Computer Science 284 (2011) 93 – 104, proceedings of The Second International Workshop on Static Analysis and Systems Biology (SASB 2011).

[20] A. Richard, Negative circuits and sustained oscillations in asynchronous automata networks, Advances in Applied Mathematics 44 (4) (2010) 378–392.

[21] C. Baral, Knowledge Representation, Reasoning and Declarative Problem Solving, Cambridge University Press, 2003.

[22] Z. Khalis, J.-P. Comet, A. Richard, G. Bernot, The SMBioNet method for discovering models of gene regulatory networks, Genes, Genomes and Genomics 3(special issue 1) (2009) 15–22.

[23] F. Corblin, E. Fanchon, L. Trilling, Applications of a formal approach to decipher discrete genetic networks, BMC Bioinformatics 11 (1) (2010) 385.

[24] F. Corblin, E. Fanchon, L. Trilling, C. Chaouiya, D. Thieffry, Automatic inference of regulatory and dynamical properties from incomplete gene interaction and expression data, in: IPCAT, Vol. 7223 of LNCS, Springer, 2012, pp. 25–30.

[25] M. Folschette, L. Paulevé, K. Inoue, M. Magnin, O. Roux, Concretizing the process hitting into biological regulatory networks, in: D. Gilbert, M. Heiner (Eds.), Computational Methods in Systems Biology, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 166–186.

[26] L. Bernardinello, F. De Cindio, A survey of basic net models and modular net classes, in: G. Rozenberg (Ed.), Advances in Petri Nets 1992, Vol. 609 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 1992, pp. 304–351.

[27] L. Paulevé, M. Magnin, O. Roux, From the Process Hitting to Petri Nets and Back, Technical Report hal-00744807 (2012).
URL `http://hal.archives-ouvertes.fr/hal-00744807`

[28] R. Thomas, R. d'Ari, Biological feedback, CRC press, 1990.

[29] G. Bernot, F. Cassez, J.-P. Comet, F. Delaplace, C. Müller, O. Roux, Semantics of biological regulatory networks, Electronic Notes in Theoretical Computer Science 180 (3) (2007) 3 – 14.

[30] M. Noual, D. Regnault, S. Sené, About non-monotony in boolean automata networks, Theoretical Computer Science 504 (2013) 12–25.

[31] E. H. Snoussi, Qualitative dynamics of piecewise-linear differential equations: a discrete mapping approach, Dynamics and stability of Systems 4 (3-4) (1989) 565–583.

[32] L. Paulevé, C. Chancellor, M. Folschette, M. Magnin, O. Roux, Logical Modeling of Biological Systems, Wiley, 2014, Ch. Analyzing Large Network Dynamics with Process Hitting, pp. 125 – 166.

[33] C. Baral, G. Gelfond, T. C. Son, E. Pontelli, Using answer set programming to model multi-agent scenarios involving agents' knowledge about other's knowledge, in: W. van der Hoek, G. A. Kaminka, Y. Lespérance, M. Luck, S. Sen (Eds.), Proceedings of the 9th International Conference

on Autonomous Agents and Multiagent Systems (AAMAS 2010), International Foundation for Autonomous Agents and Multiagent Systems, 2010, pp. 259–266.

[34] D. Thieffry, R. Thomas, Dynamical behaviour of biological regulatory networks — II. immunity control in bacteriophage lambda, Bulletin of Mathematical Biology 57 (2) (1995) 277–297.

[35] O. Sahin, H. Frohlich, C. Lobke, U. Korf, S. Burmester, M. Majety, J. Mattern, I. Schupp, C. Chaouiya, D. Thieffry, A. Poustka, S. Wiemann, T. Beissbarth, D. Arlt, Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance, BMC Systems Biology 3 (1).

[36] S. Klamt, J. Saez-Rodriguez, J. Lindquist, L. Simeoni, E. Gilles, A methodology for the structural and functional analysis of signaling and regulatory networks, BMC Bioinformatics 7 (1) (2006) 56.

[37] J. Saez-Rodriguez, L. Simeoni, J. A. Lindquist, R. Hemenway, U. Bommhardt, B. Arndt, U.-U. Haus, R. Weismantel, E. D. Gilles, S. Klamt, et al., A logical model provides insights into t cell receptor signaling, PLoS Computational Biology 3 (8) (2007) e163.

[38] R. Samaga, J. Saez-Rodriguez, L. G. Alexopoulos, P. K. Sorger, S. Klamt, The logic of egfr/erbb signaling: Theoretical properties and analysis of high-throughput data, PLoS Computational Biology 5 (8) (2009) e1000438.

[39] S. Klamt, J. Saez-Rodriguez, E. D. Gilles, Structural and functional analysis of cellular networks with cellnetanalyzer, BMC Systems Biology 1 (1) (2007) 2.

[40] G. Bernot, J.-P. Comet, Z. Khalis, Gene regulatory networks with multiplexes, in: European Simulation and Modelling Conference Proceedings, 2008, pp. 423–432.

Appendix. List of actions of the phage lambda immunity control model used in Subsect. 7.1

$$
\begin{aligned}
\mathcal{H} = \{ \quad & \mathsf{cI}_0 \to \mathsf{cI\text{-}cro}_0 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_2 && \mathsf{cI}_0 \to \mathsf{cI\text{-}cro}_1 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_3 \\
& \mathsf{cI}_0 \to \mathsf{cro}_0 \,\vec{\Gamma}\, \mathsf{cro}_1 && \mathsf{cI}_0 \to \mathsf{cro}_1 \,\vec{\Gamma}\, \mathsf{cro}_2 \\
& \mathsf{cI}_0 \to \mathsf{cro}_2 \,\vec{\Gamma}\, \mathsf{cro}_3 && \mathsf{cI}_1 \to \mathsf{cI\text{-}cro}_0 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_2 \\
& \mathsf{cI}_1 \to \mathsf{cI\text{-}cro}_1 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_3 && \mathsf{cI}_1 \to \mathsf{cI\text{-}cro}_2 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_0 \\
& \mathsf{cI}_1 \to \mathsf{cI\text{-}cro}_3 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_1 && \mathsf{cI}_1 \to \mathsf{cro}_0 \,\vec{\Gamma}\, \mathsf{cro}_1 \\
& \mathsf{cI}_1 \to \mathsf{cro}_1 \,\vec{\Gamma}\, \mathsf{cro}_2 && \mathsf{cI}_1 \to \mathsf{cro}_2 \,\vec{\Gamma}\, \mathsf{cro}_3 \\
& \mathsf{cI}_2 \to \mathsf{cI\text{-}cro}_2 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_0 && \mathsf{cI}_2 \to \mathsf{cI\text{-}cro}_3 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_1 \\
& \mathsf{cI}_2 \to \mathsf{cro}_1 \,\vec{\Gamma}\, \mathsf{cro}_0 && \mathsf{cI}_2 \to \mathsf{cro}_2 \,\vec{\Gamma}\, \mathsf{cro}_1 \\
& \mathsf{cI}_2 \to \mathsf{cro}_3 \,\vec{\Gamma}\, \mathsf{cro}_2 && \mathsf{cI\text{-}cro}_0 \to \mathsf{cI\text{-}cro\text{-}N}_2 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro\text{-}N}_0 \\
& \mathsf{cI\text{-}cro}_0 \to \mathsf{cI\text{-}cro\text{-}N}_3 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro\text{-}N}_1 && \mathsf{cI\text{-}cro}_0 \to \mathsf{N}_1 \,\vec{\Gamma}\, \mathsf{N}_0 \\
& \mathsf{cI\text{-}cro}_1 \to \mathsf{cI\text{-}cro\text{-}N}_2 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro\text{-}N}_0 && \mathsf{cI\text{-}cro}_1 \to \mathsf{cI\text{-}cro\text{-}N}_3 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro\text{-}N}_1 \\
& \mathsf{cI\text{-}cro}_1 \to \mathsf{N}_1 \,\vec{\Gamma}\, \mathsf{N}_0 && \mathsf{cI\text{-}cro}_2 \to \mathsf{cI\text{-}cro\text{-}N}_2 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro\text{-}N}_0 \\
& \mathsf{cI\text{-}cro}_2 \to \mathsf{cI\text{-}cro\text{-}N}_3 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro\text{-}N}_1 && \mathsf{cI\text{-}cro}_2 \to \mathsf{N}_1 \,\vec{\Gamma}\, \mathsf{N}_0 \\
& \mathsf{cI\text{-}cro}_3 \to \mathsf{cI\text{-}cro\text{-}N}_0 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro\text{-}N}_2 && \mathsf{cI\text{-}cro}_3 \to \mathsf{cI\text{-}cro\text{-}N}_1 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro\text{-}N}_3 \\
& \mathsf{cI\text{-}cro}_3 \to \mathsf{N}_0 \,\vec{\Gamma}\, \mathsf{N}_1 && \mathsf{cI\text{-}cro\text{-}N}_0 \to \mathsf{cII}_1 \,\vec{\Gamma}\, \mathsf{cII}_0 \\
& \mathsf{cI\text{-}cro\text{-}N}_1 \to \mathsf{cII}_1 \,\vec{\Gamma}\, \mathsf{cII}_0 && \mathsf{cI\text{-}cro\text{-}N}_2 \to \mathsf{cII}_1 \,\vec{\Gamma}\, \mathsf{cII}_0 \\
& \mathsf{cI\text{-}cro\text{-}N}_3 \to \mathsf{cII}_0 \,\vec{\Gamma}\, \mathsf{cII}_1 && \mathsf{cII}_0 \to \mathsf{cro\text{-}cII}_0 \,\vec{\Gamma}\, \mathsf{cro\text{-}cII}_1 \\
& \mathsf{cII}_0 \to \mathsf{cro\text{-}cII}_2 \,\vec{\Gamma}\, \mathsf{cro\text{-}cII}_3 && \mathsf{cII}_1 \to \mathsf{cro\text{-}cII}_1 \,\vec{\Gamma}\, \mathsf{cro\text{-}cII}_0 \\
& \mathsf{cII}_1 \to \mathsf{cro\text{-}cII}_3 \,\vec{\Gamma}\, \mathsf{cro\text{-}cII}_2 && \mathsf{cro}_0 \to \mathsf{cI\text{-}cro}_0 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_1 \\
& \mathsf{cro}_0 \to \mathsf{cI\text{-}cro}_2 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_3 && \mathsf{cro}_0 \to \mathsf{cro\text{-}cII}_2 \,\vec{\Gamma}\, \mathsf{cro\text{-}cII}_0 \\
& \mathsf{cro}_0 \to \mathsf{cro\text{-}cII}_3 \,\vec{\Gamma}\, \mathsf{cro\text{-}cII}_1 && \mathsf{cro}_1 \to \mathsf{cI\text{-}cro}_0 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_1 \\
& \mathsf{cro}_1 \to \mathsf{cI\text{-}cro}_2 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_3 && \mathsf{cro}_1 \to \mathsf{cro\text{-}cII}_0 \,\vec{\Gamma}\, \mathsf{cro\text{-}cII}_2 \\
& \mathsf{cro}_1 \to \mathsf{cro\text{-}cII}_1 \,\vec{\Gamma}\, \mathsf{cro\text{-}cII}_3 && \mathsf{cro}_2 \to \mathsf{cI\text{-}cro}_0 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_1 \\
& \mathsf{cro}_2 \to \mathsf{cI\text{-}cro}_1 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_0 && \mathsf{cro}_2 \to \mathsf{cI\text{-}cro}_2 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_3 \\
& \mathsf{cro}_2 \to \mathsf{cI\text{-}cro}_3 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_2 && \mathsf{cro}_2 \to \mathsf{cro\text{-}cII}_0 \,\vec{\Gamma}\, \mathsf{cro\text{-}cII}_2 \\
& \mathsf{cro}_2 \to \mathsf{cro\text{-}cII}_1 \,\vec{\Gamma}\, \mathsf{cro\text{-}cII}_3 && \mathsf{cro}_3 \to \mathsf{cI\text{-}cro}_1 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_0 \\
& \mathsf{cro}_3 \to \mathsf{cI\text{-}cro}_3 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro}_2 && \mathsf{cro}_3 \to \mathsf{cro}_3 \,\vec{\Gamma}\, \mathsf{cro}_2 \\
& \mathsf{cro}_3 \to \mathsf{cro\text{-}cII}_0 \,\vec{\Gamma}\, \mathsf{cro\text{-}cII}_2 && \mathsf{cro}_3 \to \mathsf{cro\text{-}cII}_1 \,\vec{\Gamma}\, \mathsf{cro\text{-}cII}_3 \\
& \mathsf{cro\text{-}cII}_0 \to \mathsf{cI}_0 \,\vec{\Gamma}\, \mathsf{cI}_1 && \mathsf{cro\text{-}cII}_0 \to \mathsf{cI}_1 \,\vec{\Gamma}\, \mathsf{cI}_2 \\
& \mathsf{cro\text{-}cII}_1 \to \mathsf{cI}_0 \,\vec{\Gamma}\, \mathsf{cI}_1 && \mathsf{cro\text{-}cII}_1 \to \mathsf{cI}_1 \,\vec{\Gamma}\, \mathsf{cI}_2 \\
& \mathsf{cro\text{-}cII}_2 \to \mathsf{cI}_0 \,\vec{\Gamma}\, \mathsf{cI}_1 && \mathsf{cro\text{-}cII}_2 \to \mathsf{cI}_1 \,\vec{\Gamma}\, \mathsf{cI}_2 \\
& \mathsf{cro\text{-}cII}_3 \to \mathsf{cI}_1 \,\vec{\Gamma}\, \mathsf{cI}_0 && \mathsf{cro\text{-}cII}_3 \to \mathsf{cI}_2 \,\vec{\Gamma}\, \mathsf{cI}_1 \\
& \mathsf{N}_0 \to \mathsf{cI\text{-}cro\text{-}N}_1 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro\text{-}N}_0 && \mathsf{N}_0 \to \mathsf{cI\text{-}cro\text{-}N}_3 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro\text{-}N}_2 \\
& \mathsf{N}_1 \to \mathsf{cI\text{-}cro\text{-}N}_0 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro\text{-}N}_1 && \mathsf{N}_1 \to \mathsf{cI\text{-}cro\text{-}N}_2 \,\vec{\Gamma}\, \mathsf{cI\text{-}cro\text{-}N}_3 \qquad \}
\end{aligned}
$$