HAL
open science

A modular, qualitative modelling of regulatory networks using Petri nets

Hanna Klaudel, Claudine Chaouiya, Franck Pommereau

To cite this version:

Hanna Klaudel, Claudine Chaouiya, Franck Pommereau. A modular, qualitative modelling of regulatory networks using Petri nets. Modeling in Systems Biology, pp.253-279, 2011, 10.1007/978-1-84996-474-6_12 . hal-02309999

HAL Id: hal-02309999

https://hal.science/hal-02309999v1

Submitted on 9 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A modular, qualitative modelling of regulatory networks using Petri nets

C. Chaouiya, H. Klaudel, F. Pommereau

Abstract Advances in high-throughput technologies have enabled the delineation of large networks of interactions that control cellular processes. To understand behavioural properties of these complex networks, mathematical and computational tools are required. The multi-valued logical formalism, initially defined by R. Thomas and co-workers, proved well adapted to account for the qualitative knowledge available on regulatory interactions, and also to perform analyses of their dynamical properties. In this context, we present two representations of logical models in terms of Petri nets. In a first step, we briefly show how logical models of regulatory networks can be transposed into standard (place/transition) Petri nets, and discuss the capabilities of such representation. In the second part, we focus on logical regulatory modules and their composition, demonstrating that a high-level Petri net representation greatly facilitates the modelling of interconnected modules. Doing so, we introduce an explicit means to integrate signals from various interconnected modules, taking into account their spatial distribution. This provides a flexible modelling framework to handle regulatory networks that operate at both intra- and intercellular levels. As an illustration, we describe a simplified model of the segment-polarity module involved in the segmentation of the *Drosophila* embryo.

C. Chaouiya
Instituto Gulbenkian de Ciência, Rua da Quinta Grande, 6 P-2780-156 Oeiras, Portugal
TAGC Inserm U928, Campus de Luminy, case 928, 13288 Marseille, France
e-mail: chaouiya@igc.gulbenkian.pt

H. Klaudel
IBISC, université d'Evry, 523 place des terrasses, 91000 Evry, France
e-mail: klaudel@ibisc.univ-evry.fr

F. Pommereau
LACL, université Paris Est, 61 avenue du général de Gaulle, 94010 Créteil, France
e-mail: pommereau@univ-paris12.fr

1 Introduction

Great advances in molecular biology, genomics and functional genomics open the way to the understanding of regulatory mechanisms controlling essential biological processes. These mechanisms interplay and operate at diverse levels (transcription and translation of the genetic material, protein modifications, etc.). They define large and complex networks, which in turn constitute a relevant functional integrative framework to study the regulation of cellular processes. To assess the behaviours induced by such networks, dedicated mathematical and computational tools are very much required. In general, mathematical models for concrete regulatory networks are defined as a unique whole, considering networks of limited sizes (up to few dozens of components). This approach is not scalable and has to be modified as networks are increasing in size and complexity. One main purpose of this chapter is to present a compact, qualitative modelling framework to represent large regulatory networks and analyse them.

We rely on a qualitative discrete framework for the modelling of regulatory networks, namely the generalised logical formalism, initially proposed by R. Thomas in the 70s [43, 44, 45]. The logical formalism has been applied to a variety of regulatory networks comprising relatively large numbers of components (e.g. [34, 35]). To tackle the modelling of networks encompassing hundreds of nodes or interacting cells, we propose here to resort to modular modelling. In particular, in the case of patterning in developmental processes, one has to consider patches of communicating cells. In such processes, modularity clearly arises, each intra-cellular network defining a module. More precisely, in this chapter, we provide a convenient way to define the modelling of interacting regulatory modules.

After defining the semantics underlying regulatory interactions (as opposed to biochemical reactions that compose e.g. metabolic networks), the Section 2 gives the basis of the logical formalism. In [6, 7, 11, 41], standard (i.e., P/T) Petri net representation of logical regulatory graphs have been proposed. This representation is summarised and discussed in Section 3.

The rest of the chapter is dedicated to the specification of a framework that addresses module composition in the context of patches of communicating cells. In Section 4, we show how, based on the logical framework, high-level Petri net representation provides a very compact and efficient means to compose regulatory modules.

Finally, to illustrate the modelling framework delineated in Section 4, we show the dynamical analyses (in particular expression pattern identification) for various composition scenarios of a simple module, and of the segment-polarity module involved in the segmentation of the *Drosophila* embryo.

2 Logical modelling of regulatory networks

Regulation refers to the molecular mechanisms responsible for the changes in the concentration or activity of a functional product. Such mechanisms range from DNA-RNA transcription to post-translational protein modifications. In regulatory networks, details on the precise molecular mechanisms that drive the regulation are often abstracted, the semantics associated to the interactions mostly reduces to activatory or inhibitory effects. Among the diversity of modelling frameworks used to model such regulatory networks (see [14, 38]), one successful qualitative formalism is the *logical* approach, initially developed by R. Thomas and co-workers [43, 44, 45]. The logical formalism has been applied to model and analyse regulatory networks controlling a variety of cellular processes from pattern formation and cell differentiation (e.g [36, 37, 35, 26]) to cell cycle (e.g. [18]). A software has been developed, GINsim, which enables the definition and analysis of logical models [20, 28] (see also Section 3.1). GINsim provides a number of exports of logical models among which several exports into Petri net formats.

When considering regulatory networks, the semantics associated with the interactions between components varies compared to that of e.g. reaction networks: levels of regulators do not change during the regulatory process. At this level of abstraction conveyed by the logical formalism, regulatory networks can be viewed as influence networks. In terms of PNs, to represent such interactions, *test arcs* provide a convenient solution. Moreover, in the case of an activation, the presence of an activator enhances the level of its target, but the absence of the activator may also have an effect on the target, decreasing its level (and the other way around for a repression). Such situations can be represented in PNs using *inhibitor arcs* that allow a test to zero. However, the analysis methods based on the matrix representation of PNs are no more valid when using inhibitor arcs. Opportunely, when places are bounded (their markings are limited), it is possible to replace inhibitor arcs by adding new complementary places. Section 3 relies on these principles to define a systematic translation of logical into Petri net models (see Part I, Chapter 3).

In the sequel, the definitions of logical regulatory graphs and their associated transition graphs are given, further details might be found in [10, 30].

2.1 Logical Regulatory Graphs (LRGs)

A Logical Regulatory Graph (LRG) is a graph, where each node represents a regulatory component, associated with a range of discrete functional levels (of expression or of activity). In most of the cases, the Boolean abstraction is sufficient (e.g. a gene is expressed or not, a protein is active or not),

but there are situations where more than two levels are necessary to convey the functional role of a regulator that varies when the concentration of its product crosses different thresholds. In particular, this is the case when a product regulates several targets, these regulations possibly occuring at distinct thresholds. This leads to the arcs of the LRG that represent regulatory interactions. Finally, one needs to define the behaviours of the components submitted to regulatory interactions. This is done by setting up logical functions that define the target levels of the components (within the admissible ranges) for each possible combinations of incoming interactions. The formal definition of LRGs is provided below, and Figure 1 page 6 gives a simple example.

Definition 1. A *logical regulatory graph* (LRG) is defined as a labelled directed multigraph[1] $\mathscr{R} = (\mathscr{G}, \mathscr{E}, \mathscr{K})$ where,

- $\mathscr{G} = \{g_1, \ldots, g_n\}$ is the set of nodes, representing *regulatory components*. Each $g_i \in \mathscr{G}$ is associated to its *maximum level* Max_i ($Max_i \in \mathbb{N}^*$), its *current level* being represented by the variable x_i ($x_i \in \{0, \ldots, Max_i\}$). We define $x \stackrel{\text{df}}{=} (x_1, \ldots, x_n)$ the current state, and $\mathscr{S} \stackrel{\text{df}}{=} \prod_{g_i \in \mathscr{G}} \{0, \ldots, Max_i\}$ the set of all possible states.
- $\mathscr{K} = (\mathscr{K}_1, \ldots, \mathscr{K}_n)$ defines the *logical functions* attached to the nodes specifying their behaviours: $\mathscr{K}_j$ is a multi-valued logical function that gives the *target level* of g_j, depending on the state of the system: $\mathscr{K}_j : \mathscr{S} \to \{0, \ldots Max_j\}$.
- $\mathscr{E}$ is the set of oriented edges (or arcs) representing *regulatory interactions*. An arc (g_i, g_j) specifies that g_i regulates g_j (when there is no possible confusion, i stands for g_i), i.e., $\mathscr{K}_j$ varies with x_i. A regulatory graph may contain self-loops (an arc (i,i) represents a self-regulation of g_i).
  For each $g_j \in \mathscr{G}$, $Reg(j)$ denotes the set of its regulators: $i \in Reg(j)$ if and only if $(i,j) \in \mathscr{E}$.

Several remarks follow from Definition 1.

Remark 1. It is clear that, to determine the target level of a component, only the levels of its regulators are required (other components have no effet). In other words: $\mathscr{K}_j$ can be defined on the restricted domain $\prod_{g_i \in Reg(j)} \{0, \ldots Max_i\}$. For example, in Figure 1, since $G1$ is the sole regulator of $G0$, we could restrict the domain of $\mathscr{K}_{G0}$ to $\{0, \ldots Max_{G1}\}$ (indeed, in the table defining $\mathscr{K}_{G0}$, we can verify that the values of $G0$, $G2$ and $G3$ do not matter).

Remark 2. If $g_i \in Reg(j)$ and $Max_i > 1$ (g_i regulates g_j and is multi-valued), g_i may have different effects onto a component g_j, depending on the current level of g_i, leading to the definition of a multi-arc between g_i and g_j. Such a situation typically happens when a component has a dual regulatory role, e.g. activation at low and repression at high concentrations.

[1] A multigraph is a graph with possibly several edges between a pair of nodes.

Here, to avoid the cumbersome notations resulting from such multi-arcs, we assume that all interactions are simple. However, it is straightforward to generalise all the definitions introduced in this chapter to LRGs encompassing multi-arcs (see e.g. [30]).

Remark 3. The biologists often associate signs to the regulatory interactions, distinguishing between positive effect (activation or enhancing) and negative effect (repression or silencing). However the actual effect of an interaction on its target may depend on the presence of co-factors; its sign may even change depending on the context. In any case, the signs of interactions can be derived from the logical functions $\mathscr{K}$s. Moreover, a *threshold* θ associated to an interaction from g_j to g_i with $1 \leq \theta \leq Max_j$ indicates for which level of g_j the interaction is *active* (i.e., when $x_j \geq \theta$, (g_j, g_i) is active). This threshold can also be recovered from the function $\mathscr{K}_i$: θ is the value for which there exists $x \in \mathscr{S}$ such that for x' defined as $x'_k \stackrel{\text{df}}{=} x_k$, $\forall k \neq j$ and $x'_j \stackrel{\text{df}}{=} x_j + 1 = \theta$, we have $\mathscr{K}_i(x) \neq \mathscr{K}_i(x')$. For example, in Figure 1 the interaction from $G0$ to $G1$ is associated with a threshold 1 and a positive sign; this is visible by comparing the first and third rows of the table defining $\mathscr{K}_{G2}$ (also the second and the fourth rows), where, for fixed values of the other components, changing $G0$ level from 0 to 1 leads to a change in the target value of $G1$ from 0 to 1. Whereas the interaction from $G0$ to $G3$ is associated with a threshold 2 with a negative sign (determined by comparing second and third lines of $\mathscr{K}_{G3}$ table).

Finally, it is worth noting that a set of logical functions $\mathscr{K}_i, i = \{1, \ldots n\}$, fully defines an LRG encompassing n regulatory components. In particular, for each $i \in \{1, \ldots n\}$, its maximum level is given by the maximum value of $\mathscr{K}_i$.

2.2 Logical functions representation based on decision diagrams

In [30], logical functions were represented by means of Reduced Ordered Multi-valued Decision Diagrams (ROMDD, referred to as MDD in the following). This representation, internally used in GINsim for efficiency purposes, facilitates the definition of algorithms for the analysis of logical models (e.g. see the stable state determination described in [30]). The use of MDDs also makes the definition of the P/T net representation of logical models easier and more consise than that proposed in e.g. [6]. This representation, which is quite intuitive, is informally presented below.

Binary Decision Diagrams (BDD) are a convenient data structure to represent Boolean functions [5]. A BDD is a rooted, directed, acyclic graph, encompassing *decision nodes* (labelled by a Boolean variable) and two leaves (also called terminal-nodes) labelled respectively 0 (false) and 1 (true). Decision nodes have two successors (or children): the left (resp. right) outgoing

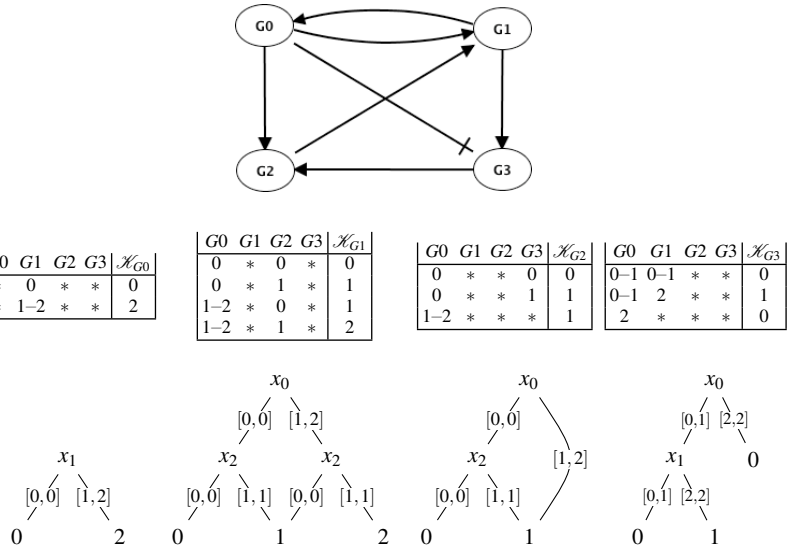


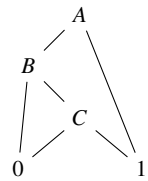
Fig. 1 Example of an LRG. The regulatory graph is displayed in the higher level, with the nodes denoting components and the arcs denoting interactions (arcs with normal arrows denote activations whereas arcs with blunt end denote repressions). The logical functions are then given in the form of tables (one for each component), where each row corresponds to (a set of) state(s) with the corresponding function values ($*$ denotes one value among all possible values, 1–2 denotes value 1 or 2, etc.). For example, the table on the left defines the logical function $\mathscr{K}_{G0}$, which only depends on the levels of $G1$, the sole regulator of $G0$. The MDD representation of each logical function is given in the lower level (see Section 2.2 for explanations). For instance, $G3$ has two regulators ($G0$ and $G1$), its logical function $\mathscr{K}_{G3}$ specifies that, when both regulator levels are lower than 2, whatever the levels of the other components (which have indeed no effect on $G3$), the target level of $G3$ is 0 (first row of the table defining $\mathscr{K}_{G3}$. The MDD representing $\mathscr{K}_{G3}$ encompasses the decision variables x_0 and x_1 (the levels of $G3$ regulators) and the case just described is recovered by following the branchs going out x_0 and x_1 labelled $[0,1]$, which leads to a leaf labelled 0 (for readability, this leaf has been duplicated).

edge represents an assignment of the variable to 0 (resp. to 1). Most of BDD are ordered and reduced, meaning that variables appear in the same order along all the paths from the root, and that all isomorphic (equal) subgraphs are merged and nodes having two isomorphic children are deleted (see Figure 2 for an illustration). This structure has been naturally extended to handle discrete multi-valued functions, which are then represented by Multi-valued Decision Diagrams (MDD), where the decision nodes have as many children as the number of their possible values and the leaves (or terminal nodes) are labelled by the values of the function [23]. The ordering and reduction rules defined for binary decision diagrams apply also to this multi-valued generalisation (see [5] for further details). A path from the root to a leaf represents a

(possibly partial) assignment of the decision variables for which the function takes the value carried by the leaf (see Figure 2).

$$f_1(A,B,C) = A \vee (B \wedge C)$$

A	B	C	f_1
0	0	*	0
0	*	0	0
0	1	1	1
1	*	*	1



$$f_2(A,B) = \begin{cases} 0 \ if \ (A=0) \wedge (B=0) \\ 1 \ if \ ((A=0) \wedge (B=1)) \vee (A=1) \\ \quad \vee ((A=2) \wedge (B=0)) \\ 2 \ if \ (A=2) \wedge (B=1) \end{cases}$$

A	B	f_2
0	0	0
0	1	1
1	*	1
2	0	1
2	1	2

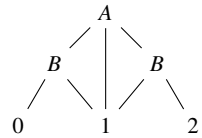


Fig. 2 Decision diagrams representing logical functions. The Boolean function f_1 (top part) of the Boolean variables A,B,C, evaluates 1 (i.e. true) if A or B and C ($\wedge$ stands for and, $\vee$ stands for or); f_1 can be equivalently defined by a truth table (where $*$ denotes all possible values, here 0 and 1) and by a binary decision diagram. For instance, A,B,C all set to 0 (first line of the table) corresponds to the path in the diagram following the leftmost edges going out A and B and leading to the leaf 0, indicating that, for these assignments of A,B,C, the function f_1 evaluates to 0. The multi-valued function f_2 (bottom part) evaluates to 0, 1 or 2, depending on the values of the discrete variables A and B. Similarly, f_2 can be defined by means of a table or a decision diagram. For instance, in this diagram, the variable assignments $A=2$ and $B=1$ (last line of the table) define the path comprising the rightmost edge going out A followed by the righmost edge going out B, leading to the value 2.

In the MDD representation of the function $\mathscr{K}_i$ of a regulatory component g_i, the decision variables are the variables associated to the regulators of g_i, and the leaves take their values in $[0, Max_i]$. Note that the use of MDD leads to a simplified expression of $\mathscr{K}_i$, but the resulting diagram and its complexity may vary depending on the ordering of the decision variables [23].

Finally, it is possible to consider a compact representation of the usual MDD by merging consecutive edges leading to the same child as explained hereafter.

Given $g_i \in \mathscr{G}$, for each decision variable x_j that appears in the diagram of $\mathscr{K}_i$, there are $Max_j + 1$ outgoing edges, implicitly labelled with the corresponding value in $[0, Max_j]$. An alternative diagram can thus be considered by merging consecutive edges (i.e. labelled with consecutive values) towards the same child into a unique edge, which is then labelled with the integer interval of these consecutive values. Remaining edges are labelled by intervals containing a unique value for the decision variable. Then, in the resulting diagram, each decision path Φ (from the root node to a leaf labelled $v_\Phi \in [0, Max_i]$) corresponds to a set of assignments of the regulators $g_j \in Reg(i)$ for which the value of $\mathscr{K}_i$ is v_Φ (see Exercise 1):

- If path Φ encompasses an edge going out the decision variable x_j, the set of assignments of x_j is equal to the label $[\phi_j, \phi'_j] \subsetneq [0, Max_j]$ (called the Φ assignment interval for x_j) of the edge going out the decision variable x_j.
- If, along the path Φ, a decision variable x_j does not appear (due to the simplication of the MDD), it means that $\mathscr{K}_i(x) = v_\Phi$ does not depend on x_j.

Such diagrams were introduced in [42] as a generalisation of MDD and refered to as Interval Decision Diagrams (IDD). Here, we deal with a specific class of IDD, since our decision variables are discrete and bounded.

Remark that, because we have assumed that regulatory graphs do not encompass multi-arcs, in this IDD representation, each decision variable has exactly two children.

Figure 1 illustrates an LRG together with the functions $\mathscr{K}$s represented by means of IDDs. For a better readability, IDD are often not fully reduced.

2.3 State Transition Graphs associated to LRGs

The behaviour of a logical regulatory graph is defined by the logical functions introduced in Definition 1. For any state of the system (i.e. a vector encompassing the levels of all the regulatory components), these functions indicate the target level of each component, that is the level to which it evolves. State Transition Graphs (STGs) constitute a classical and convenient way of representing the behaviour of such systems. In these directed graphs, nodes represent states, and arcs represent transitions between states that amount to update one component level (increasing or decreasing it by one).

Definition 2. Given an LRG $\mathscr{R} = (\mathscr{G}, \mathscr{E}, \mathscr{K})$, its full *State Transition Graph (STG)* is a directed graph $(\mathscr{S}, \mathscr{T})$ such that:

- the set of nodes is the set of states $\mathscr{S}$ as defined above,
- $\mathscr{T} \subseteq \mathscr{S}^2$ defines the set of transitions (arcs) as follows. For all $(x, y) \in \mathscr{S}^2$:

$$(x,y) \in \mathscr{T} \Leftrightarrow \exists g_i \in \mathscr{G} \text{ s.t.} \begin{cases} \mathscr{K}_i(x) \neq x_i, \\ y_i = \delta_i(x) \stackrel{\text{df}}{=} x_i + sign(\mathscr{K}_i(x) - x_i), \\ y_j = x_j, \quad \forall j \neq i. \end{cases}$$

Given an initial state x^0, we further define the STG $(\mathscr{S}_{|x^0}, \mathscr{T}_{|x^0})$, which contains all states reachable from x^0:

- $x^0 \in \mathscr{S}_{|x^0}$,
- $\forall x \in \mathscr{S}_{|x^0}$, $\forall y \in \mathscr{S}$, $(x,y) \in \mathscr{T} \Rightarrow y \in \mathscr{S}_{|x^0}$ and $(x,y) \in \mathscr{T}_{|x^0}$.

The *updating function* δ_i of a regulatory component g_i, as defined above, specifies the update of x_i, the level of g_i; depending on the current state x

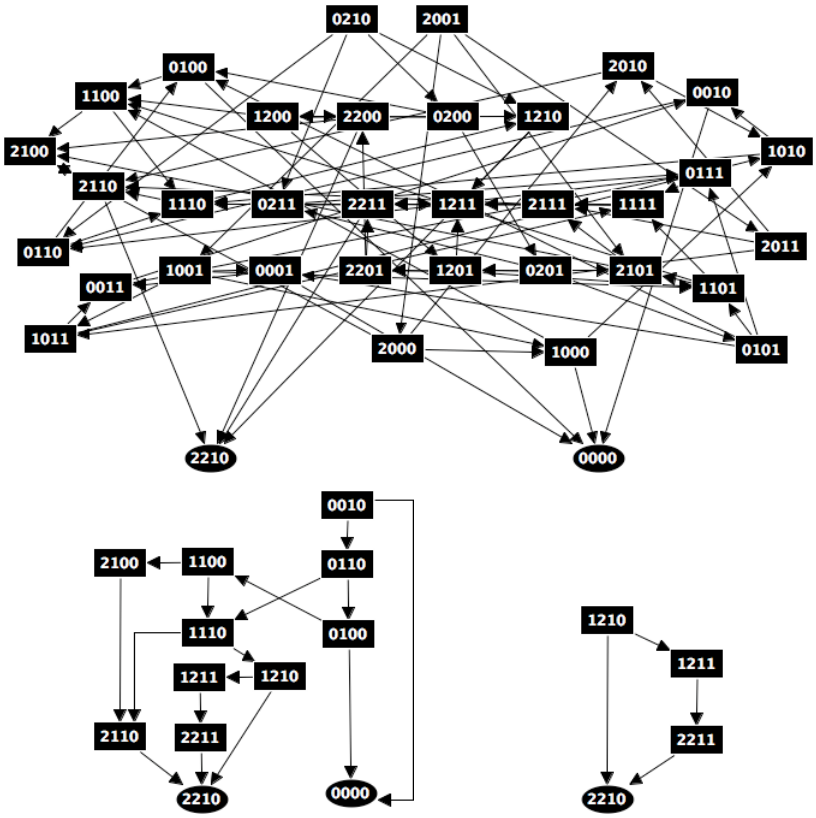


Fig. 3 State transition graphs corresponding to the LRG of Figure 1. On top, the full STG, encompassing all 36 states. Note that there are two stable states indicated as ellipse nodes: $(x_{G0}, x_{G1}, x_{G2}, x_{G3}) = (2,2,1,0)$ and $(0,0,0,0)$. Other transient states are denoted as rectangular nodes. The two graphs in the lower part of the figure correspond to sub-graphs of the full STG for the initial states $(0,0,1,0)$ (on the left) and $(1,2,1,0)$ (on the right). Note that, for the initial state $(1,2,1,0)$, one stable state is lost.

of the network, the value of $\delta_i(x)$ is either x_i (no change), $x_i + 1$ (increase by one) or $x_i - 1$ (decrease by one).

Analysing the STG, we can recover important properties of an LRG, among which:

- Single point attractors or stable states (e.g. corresponding to stable expression patterns) are nodes of the STG with no successor; in other words, they are states in which all component levels are equal to the target value indicated by the logical functions.
- Complex attractors (e.g. corresponding to oscillatory behaviours) are terminal strongly connected components encompassing more than one node,

i.e. sets such that all nodes are reachable from each other along directed paths, and there is no outgoing transition; in other words, the system is trapped in such a set once it has reached one of its state (see Exercice 2).
- Reachability of given attractors from initial conditions corresponds to the existence of path(s) in the STG.

3 P/T Petri net representation

The Definition 3 below explicitly defines a P/T net associated to an LRG, using the MDD representation of the functions $\mathscr{K}_i$s (as introduced in Section 2.1). Further details, basic properties and applications of this P/T net representation of LRGs are provided in [6] for the Boolean case, and in [7, 9] for the multi-valued one.

Definition 3. Given an LRG $\mathscr{R} = (\mathscr{G}, \mathscr{M}ax, \mathscr{E}, \mathscr{K})$, we define the corresponding *Multi-valued Regulatory Petri Net* (MRPN) as follows:

- For each $g_i \in \mathscr{G}$, two complementary places are defined, g_i and $\widetilde{g_i}$, satisfying, for all marking M:
$$M(g_i) + M(\widetilde{g_i}) = Max_i. \tag{1}$$
- For each $g_i \in \mathscr{G}$, for each path Φ from the root to a leaf of the MDD representing $\mathscr{K}_i$, at most two transitions are defined, one accounting for the increasing shift (denoted $t^+_{i,\Phi}$), the second accounting for the decreasing shift (denoted $t^-_{i,\Phi}$) (this simplifies when the leaf is associated with an extreme value, see below). Recall that Φ defines assignment intervals of the levels of g_j in $Reg(i)$: $x_j \in [\phi_j, \phi'_j]$, with $\phi_j, \phi'_j \in [0, Max_j]$ and $\phi_j \leq \phi'_j$.
- Transitions $t^+_{i,\Phi}$ and $t^-_{i,\Phi}$ are connected to:
  - place g_j, $j \in Reg(i)$, with a test arc weighted ϕ_j,
  - place $\widetilde{g_j}$, $j \in Reg(i)$, with a test arc weighted $Max_j - \phi'_j$.

  Transition $t^+_{i,\Phi}$ is further connected to:
  - place g_i, with an outgoing arc (increasing the level of g_i),
  - place $\widetilde{g_i}$, with an incoming arc weighted $Max_i - v_\Phi + 1$ (ensuring that the current level of g_i is less than the focal value v_Φ) and an outgoing arc weighted $Max_i - v_\Phi$ (accounting for the decreasing by one of the current marking of this complementary place).

  Symmetrically, transition $t^-_{i,\Phi}$ is further connected to:
  - place $\widetilde{g_i}$, with an outgoing arc (decreasing the level of g_i),
  - place g_i, with an incoming arc weighted $v_\Phi + 1$ (ensuring that the current level of g_i is greater than the focal value v_Φ) and an outgoing arc weighted v_Φ (accounting for the decreasing by one of the current marking).

From the definition above, it follows that, for all $g_i \in \mathscr{G}$ and Φ a path in the decision diagram associated to $\mathscr{K}_i$, when $v_\Phi = 0$ or $v_\Phi = Max_i$ (the value of the $\mathscr{K}_i$ for this assignment of the regulators is *extreme*), only one transition is relevant. Indeed, if $v_\Phi = 0$, transition $t^+_{i,\Phi}$ can be omitted as, by construction, there will never be $Max_i + 1$ tokens in place $\widetilde{g_i}$. Similarly, if $v_\Phi = Max_i$, transition $t^-_{i,\Phi}$ can be omitted as there will never be $Max_i + 1$ tokens in place g_i. Moreover, for $g_j \in Reg(i)$, when $\phi_j = \phi'_j$, from Equation 1, it suffices to consider only one test arc (that towards place g_j for example).

Given an LRG $\mathscr{R}$ and an MDD representation of its logical functions, the Definition 3 uniquely specifies a P/T net. It can be shown that, given an initial state x^0, the STG $(\mathscr{S}_{|x^0}, \mathscr{T}_{|x^0})$ is isomorphic to the marking graph of the P/T net with the initial marking defined as $M_0(g_i) = x^0_i$ and $M_0(\widetilde{g_i}) = 1 - x^0_i$, for all components g_i (see proof in [9]). Hence, properties of an LRG can be derived from the analysis of the marking graph of its P/T net representation.

The MDD representation of the logical function leads to more compact Petri nets compared to those obtained by using decision trees or truth table representations (as in [7]). Different orderings of the variables in the MDD may generate different reductions. However, although the number of transitions may vary, it can be proved that the resulting dynamics (the marking graphs) are identical [9].

Figure 4 illustrates this P/T net representation for the LRG of the Figure 1.

3.1 Tools support

GINsim is a software dedicated to the definition and analyse of regulatory networks logical models [20, 28]. With the representation presented in Section 3 we can also employ PN tools to analyse (larger) LRGs. GINsim has been equipped with export functionalities generating files in the INA [21] format, PNML (Petri net markup language) [31] and APNN (Abstract Petri Net Notation) [4].

In the case of the analysis of the segment-polarity system as defined in [35], the reachability analysis of the expected patterns was performed using INA. We have also assessed the high complexity of the STG of this model, using tools such as the model-checking tool DSSZ-MC [8] and Tina [46].

3.2 Related works

Related works include the representation of logical regulatory networks by means of high-level Petri Nets. In [7] a high-level PN representation of LRGs is defined, encompassing one place and one transition for each regulatory component. The even more compact representation defined in [13] enables

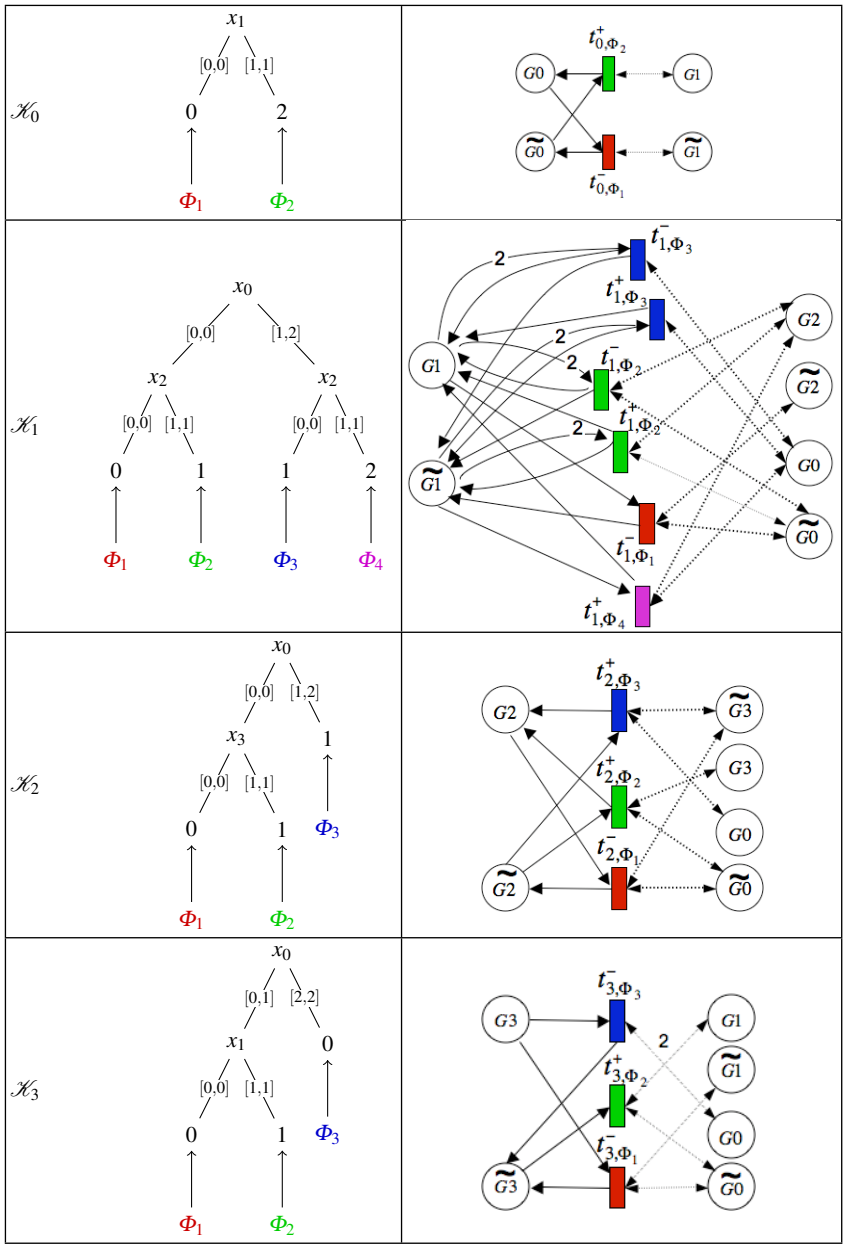


Fig. 4 P/T net representation for the LRG of the Figure 1. In each row, the MDD giving the node logical function in displayed on the left, and the corresponding piece of P/T net is displayed on the right. Test arcs are depicted as dotted lines. Paths in the MDD are indicated with their respective transition(s). For example, for $G0$, path Φ_1 (in red) corresponds to the situation where the absence of $G1$ leads to a decrease of $G0$'s value, represented in the P/T net by the transition in red t^-_{0,Φ_1}. For G_1 path Φ_3 gives rise to the two transitions t^-_{1,Φ_3} and t^+_{1,Φ_3} (in blue), since it leads to the intermediate value 1.

the verification of the model coherence under various hypotheses (accounting for observed biological behaviours such as homeostasis, multistationarity, or even more specific temporal properties).

Based on similar principles as exposed in this section, Steggles *et al.* have defined a Petri net representation of Boolean regulatory graphs [41]. However, to account for Boolean networks as introduced by S. Kauffman [24] which are synchronously updated (meaning that all components are updated at once), a two-phase method is defined to ensure the synchronisation of the updates. The first phase identifies all components that are called to change their values, the second phase performs this update synchronously.

Finally, the PN representation of regulatory networks as presented here, enables the delineation of integrated models of regulated metabolic pathways, considering a logical model of the level (a PN representation) linked to a classical PN model of the metabolic part. In [40], this approach is illustrated with a qualitative modelling of the biosynthesis of tryptophan (Trp) in E. coli.

4 Modules, their composition and high-level Petri net representation

The framework presented in this section deals with collections of spatially distributed abstract modules (non-nested) which may be cells, compartments in cells or any subsystems that one wants to model separately. Each such module is defined by a regulatory network with identified inputs. The regulation functions take into account spatial configuration between modules.

In a first step, we introduce Logical Regulatory Modules (LRMs) as LRGs equipped with external input nodes and arcs. Then, we define how such modules can be spatially located and interconnected in order to form a Collection of interconnected Logical Regulatory Modules (CLRMs). Finally, given a collection of logical regulatory modules, we define its high-level Petri net representation.

4.1 Interconnecting logical regulatory modules

In what follows, a *Logical Regulatory Module* is defined as a uniquely identified logical regulatory network associated with a set of *input* components that regulate internal ones. Notice that no output nodes are defined because any internal component may generate an external signal towards other modules; moreover, input components are not effective regulatory components in that they do not introduce any intermediary step in the signalling. These inputs are meant to combine (or integrate) external signals from other modules. Functions σ in Definition 4 perform such combinations by calculating the

levels of the integrated signals, depending on the levels of the corresponding individual incoming signals and on their attributed weights. These weights, defined as real numbers on the interval $[0;1]$, encode neighbouring relations, which in turn are defined when interconnecting the modules (see Definition 5). Hence, at this stage, in order to define a logical regulatory module, one has to specify the components that are likely to signal the module, and how input signals are combined through the functions σ. If needed, both constraints could be relaxed. In particular, postponing the specification of functions σ to the actual connexion of modules would be a straightforward extension of the current framework.

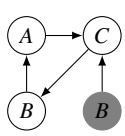
Definition 4. Given Γ a domain of regulatory components, a *Logical Regulatory Module (LRM)* $\mathscr{M}$ is a tuple $(\mathscr{G},\mathscr{E},\gamma,\sigma,\mathscr{K})$, where

- $\mathscr{G} \subseteq \Gamma$ is the set of *internal components*;
- $\mathscr{E} \subseteq \mathscr{G} \times \mathscr{G}$ is the set of *internal interactions* between internal components;
- $\gamma \subseteq \Gamma$ is the set of *input components*;
- $\sigma = \{\sigma_{v,g} \mid (v,g) \in E \subseteq \gamma \times \mathscr{G}\}$ is a set of *integration functions*,

  $$\sigma_{v,g} : (\{0,\ldots,Max_v\} \times]0;1])^* \to \{0,\ldots,Max_g\}$$

  computing the combined level for all inputs v regulating g. Their arguments are pairs (x_v,d_v) where each x_v is the level of v in one neighbour, weighted by $d_v > 0$;
- $\mathscr{K}$ is a set of logical functions defined on $\mathscr{G}$ giving, for each component $g \in \mathscr{G}$, its target level in $\{0,\ldots,Max_g\}$, depending on the levels of its regulators. For an internal regulator r ($r \in \mathscr{G}$) the level used to evaluate $\mathscr{K}_g$ is x_r as usual, while for an input regulator v ($v \in \gamma$), the level used to evaluate $\mathscr{K}_g$ is the current value of $\sigma_{v,g}$. We denote by $\mathit{args}(\mathscr{K}_g)$ the set of arguments of $\mathscr{K}_g$.

Figure 5 provides a simple example of an LRM. If $\sigma_{v,g} \in \mathit{args}(\mathscr{K}_g)$ then, v is an input regulator of g; in the pictures, this is denoted by an v-labelled arc toward the node g.



$$\mathscr{K}_A(x_B) \stackrel{\text{df}}{=} 1 \text{ if } (x_B = 1), \text{ else } 0$$
$$\mathscr{K}_B(x_C) \stackrel{\text{df}}{=} 1 \text{ if } (x_C = 1), \text{ else } 0$$
$$\mathscr{K}_C(x_A,\sigma_{B,C}) \stackrel{\text{df}}{=} 1 \text{ if } (x_A = 1 \wedge \sigma_{B,C} = 0), \text{ else } 0$$

Fig. 5 A "toy" LRM $\mathscr{M}$ with: $\mathscr{G} \stackrel{\text{df}}{=} \{A,B,C\}$, $\mathscr{E} \stackrel{\text{df}}{=} \{(A,C),(C,B),(B,A)\}$, $\gamma \stackrel{\text{df}}{=} \{B\}$, with, e.g., $\sigma_{B,C} \stackrel{\text{df}}{=} (y_j,d_j)_{j\geq 0} \mapsto \mathit{round}(\mathit{max}(y_j \cdot d_j))$, and $\mathscr{K}$ being defined on the right part of the figure. The input node B is depicted as a gray node. We have $\mathit{args}(\mathscr{K}_C) = \{A,\sigma_{B,C}\}$.

An LRM can be viewed as an *encapsulated* regulatory network with input nodes behaving as integrators of external signals of the corresponding

components from other modules. For example, for the LRM depicted in Figure 5, the level of external signal corresponding to B will be calculated as a weighted maximum over all the levels of B in neighbouring modules (i.e. having a non-zero weight). Hence, this level can be evaluated only when the module is connected to other modules (see Definition 5). Nevertheless, at this stage, it is possible to recover a fully defined LRG, by setting the integration functions and specifying the behaviours of the input components (for example as having constant levels).

Notice that, like in the Figure 5, it is not required that $\mathscr{G} \cap \gamma = \emptyset$. For example, we may have $g \in \mathscr{G} \cap \gamma$ when the regulatory component g has both autocrine (acting on the same cell) and paracrine effects (acting on neighbouring cells). Then $g \in \mathscr{G}$ accounts for the autocrine effect and $g \in \gamma$ accounts for the paracrine effect (as in the case of Wingless in the *Drosophila* segment polarity module depicted in Section 5).

We now proceed with collections of LRMs, which contain all the information needed to interconnect LRMs through their input nodes. It simply consists in defining a set of LRMs and a topological relation between these LRMs that establishes the actual connexions between modules and allows the evaluation of the levels of input components.

Definition 5. A *Collection of interconnected Logical Regulatory Module* (or CLRM) is defined as a triplet $(\mathbb{I}, \mathbb{M}, \mathbb{T})$, where:

- $\mathbb{I} \subset \mathbb{N}$ is a finite set of integers (module identifiers);
- $\mathbb{M} = \{(m, \mathscr{M}) \mid m \in \mathbb{I}\}$ is a set of LRMs, each being associates to an identifier in $\mathbb{I}$;
- $\mathbb{T} : \mathbb{I} \times \mathbb{I} \setminus \{(m,m) \mid m \in \mathbb{I}\} \to [0;1]$, is a topological (or neighbouring) relation between modules in $\mathbb{M}$; the values of $\mathbb{T}$ can be interpreted as weights associated to external signals.

In a CLRM, one can define several copies of the same LRM (these copies are distinguished by their identifiers). This is the case in Figure 6 that shows a CLRM encompassing three times the LRM of Figure 5. Notice the dashed arcs that represent how each LRM signals its neighbours according to the topological relation.

Given a CLRM, the integration functions are expressible as logical terms. Indeed, the topological relations are fixed and the values of the σ functions depend on discrete bounded variables and take discrete bounded values. Hence, it is possible to recover an LRG (a logical model for the whole set of modules). For example, consider the CLRM in Figure 6, its associated LRG is shown in Figure 7.

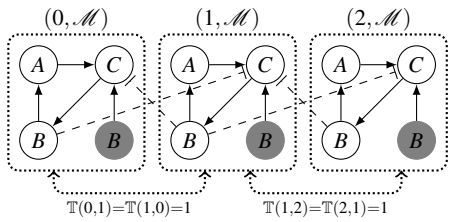
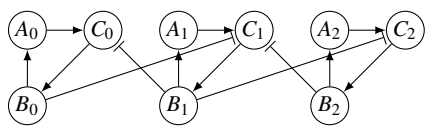


Fig. 6 A CLRM $(\mathbb{I},\mathbb{M},\mathbb{T})$ encompassing three copies of the LRM $\mathscr{M}$ of Figure 5, where: $\mathbb{I}=\{0,1,2\}$, $\mathbb{M}\stackrel{\text{df}}{=}\{(0,\mathscr{M}),(1,\mathscr{M}),(2,\mathscr{M})\}$, and $\mathbb{T}\stackrel{\text{df}}{=}\{(0,1)\mapsto 1;(1,2)\mapsto 1;(2,0)\mapsto 0\}$ completed to symmetry. Function $\sigma_{B,C}$ in module 0 depends on the level of B in module 1 (the sole connected module containing B) while $\sigma_{B,C}$ in module 1 depends on the levels of B in both modules 0 and 2 (which is depicted using dashed arcs).



$$\sigma^0_{B,C} \stackrel{\text{df}}{=} \sigma_{B,C}((x_{B_1},1))$$
$$\sigma^1_{B,C} \stackrel{\text{df}}{=} \sigma_{B,C}((x_{B_0},1),(x_{B_2},1))$$
$$\sigma^2_{B,C} \stackrel{\text{df}}{=} \sigma_{B,C}((x_{B_1},1))$$

$$\mathscr{K}_{C_0}(x_{A_0},x_{B_1}) \stackrel{\text{df}}{=} 1 \text{ if } (x_{A_0}=1\wedge x_{B_1}=0),\ \text{else } 0$$
$$\mathscr{K}_{C_1}(x_{A_1},x_{B_0},x_{B_2}) \stackrel{\text{df}}{=} 1 \text{ if } (x_{A_1}=1\wedge x_{B_0}=0\wedge x_{B_2}=0),\ \text{else } 0$$
$$\mathscr{K}'_{C_1}(x_{A_1},x_{B_0},x_{B_2}) \stackrel{\text{df}}{=} 1 \text{ if } (x_{A_1}=1\wedge (x_{B_0}=0\vee x_{B_2}=0)),\ \text{else } 0$$

Fig. 7 The LRG obtained from the CLRM of Figure 6 with integration functions defined on the right and logical functions on the bottom. If we choose $\sigma^m_{B,C}$ as the maximal value of its argument levels ($\sigma^m_{B,C}=max(x_{B_n}\cdot\mathbb{T}(n,m))$) as suggested in the caption of Figure 5, then, the logical function of C_0 is $\mathscr{K}_{C_0}$, and the logical function of C_1 is $\mathscr{K}_{C_1}$. Now, if $\sigma^m_{B,C}=min(x_{B_n}\cdot\mathbb{T}(n,m))$, then, $\mathscr{K}_{C_0}$ remains the same, but $\mathscr{K}_{C_1}$ is replaced by $\mathscr{K}'_{C_1}$. Notice that $\mathscr{K}_{C_2}$ is similar to $\mathscr{K}_{C_0}$ (replace x_{A_0} with x_{A_2}), moreover, $\mathscr{K}_{A_i}$ and $\mathscr{K}_{B_i}$ for $0\le i\le 2$ are exactly as in Figure 5.

4.2 High-level Petri net representation

We first define a class of high-level Petri nets [25] especially crafted to our needs.

Intuitively, the main difference between such high-level Petri nets and the usual P/T Petri nets is that now places may carry tokens that have a value. In our particular case, these values will be pairs (i,v) where i is the identifier of a module and v the level of one of the components of this module.

In order to consume and produce tokens when a transition fires, arcs are labelled with expressions involving variables; an empty label denotes the absence of arc. At fire time, it is necessary to *bind* (i.e., to map) each such variable to a concrete value so that the annotation on each arc can be evaluated to a collection of tokens. A *binding* β is a function that maps variables

to values. We denote $\beta(expr)$ the evaluation of an expression *expr* under the binding β.

More precisely, a *high-level Petri net* is a tuple (S,T,ℓ), where:

- S is the set of places, each place is allowed to carry structured tokens from $\mathbb{N}\times\mathbb{N}$;
- T, disjoint from S, is the set of transitions;
- $\ell : (S\times T)\cup(T\times S)$ defines the labelling of the arcs by expressions.

In general, a *marking* M of a high-level Petri net is a mapping associating to each place $s\in S$ a multiset over $\mathbb{N}\times\mathbb{N}$ representing the tokens held by s at M. However, in our case, markings are always sets (in all evolutions, no token may be duplicated in the same place). So, we assume that arc labels always evaluate to sets of tokens.

A transition t may *fire* at a marking M with binding β if there are enough tokens in the input places of t: for all $s\in S$, $\beta(\ell(s,t))\subseteq M(s)$. If so, t may fire and produce a new marking M', where M' is M in which consumed tokens are removed and produced tokens are added, i.e., for all $s\in S$: $M'(s)\stackrel{\text{df}}{=}\big(M(s)\setminus\beta(\ell(s,t))\big)\cup\beta(\ell(t,s))$, see Figure 8.

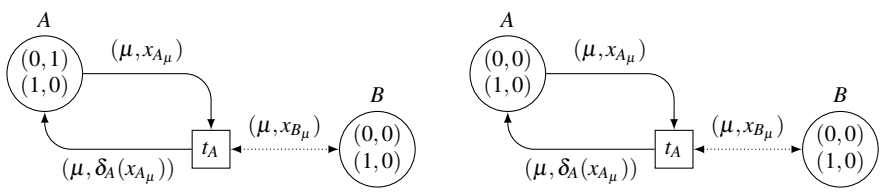


Fig. 8 The marking depicted on the left represents the fact that, for module 0, $x_A=1$ and $x_B=0$, and, for module 1, $x_A=0=x_B$. For this marking, transition t_A is enabled for the binding $\beta=(\mu\to 0, x_{A_\mu}\to 1, x_{B_\mu}\to 0)$. Suppose $\delta_A(1)=0$ (δ_A being the updating function), the firing of t_A leads to the marking depicted on the right of the picture. This correspond to a state change in the biological system.

Remark 4. For the sake of simplicity, Definition 6 below is valid only if modules do not overlap, i.e., for $(m,\mathscr{M})$ and $(n,\mathscr{N})$ two LRMs in $\mathbb{M}$, $(\mathscr{M}\neq\mathscr{N})\Rightarrow(\mathscr{G}^m\cap\mathscr{G}^n=\emptyset)$. This means that the set of all regulatory components is partitioned and we can focus on the evolution of each species in the context of the set of (identical) modules it belongs to.

Let $\mathscr{K}_g$ be the logical function of g. As previously, we define the updating function $\delta_g(x_{g_m})\stackrel{\text{df}}{=}x_{g_m}+sign(\mathscr{K}_g(\cdots)-x_{g_m})$ where $\mathscr{K}_g$ is computed with the appropriate arguments (including integration functions calls) as defined above.

Each internal regulatory component $g\in\mathscr{G}$ is modelled by a high-level Petri net place s_g that stores the numeric value of the corresponding level

for each module m. In order to model several modules (each having a unique identifier), each place holds tokens of the form (m,x_g), where $m \in \mathbb{I}$ and x_g is the level of g in module m. The evolution of each regulatory component $g \in \mathscr{G}$ is implemented by a unique transition t_g that consumes the value of g in m (token (m,x_g) from place s_g), reads the values of the regulators of g, and produces the new level $\delta_g(x_{g_m})$ of g in m. Thus, transition t_g has all the necessary arcs to the places modelling g and its regulators.

Definition 6. Let $(\mathbb{I},\mathbb{M},\mathbb{T})$ be a CLRM. We denote $\mathscr{G} \stackrel{\text{df}}{=} \bigcup_{m\in\mathbb{I}} \mathscr{G}^m$ the set of components involved in the CLRM. The high-level Petri net associated to $(\mathbb{I},\mathbb{M},\mathbb{T})$ is (S,T,ℓ), defined as:

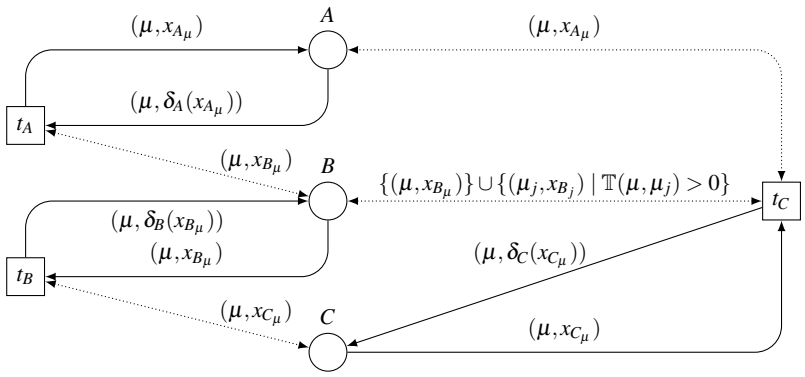
- $S \stackrel{\text{df}}{=} \{s_g \mid g \in \mathscr{G}\}$
- $T \stackrel{\text{df}}{=} \{t_g \mid g \in \mathscr{G}\}$;
- For each $g \in \mathscr{G}$, let m denotes the LRM such that $g \in \mathscr{G}^m$, the arcs attached to each $t_g \in T$ are as follows:
  - First a pair of arcs allows to read and update the current level of g for module $\mathscr{M}$, whose identifier m is captured by a net variable μ:
    - if $\sigma_{g,g} \notin \mathit{args}(\mathscr{K}_g)$, there is one arc (s_g,t_g) labelled by $\{(\mu,x_{g_\mu})\}$ and one arc (t_g,s_g) labelled by $\{(\mu,\delta_g(x_{g_\mu}))\}$;
    - otherwise, there is one arc (s_g,t_g) labelled by $\{(\mu,x_{g_\mu})\} \cup \{(\eta,x_{g_\eta}) \mid \forall\eta : \mathbb{T}(\mu,\eta) > 0\}$ and one arc (t_g,s_g) labelled by $\{(\mu,\delta_g(x_{g_\mu}))\} \cup \{(\eta,x_{g_\eta}) \mid \forall\eta : \mathbb{T}(\mu,\eta) > 0\}$; it means that the levels of g in all modules η in the neighbouring of m are read and only the level of g in m is updated;
  - Then, for each $g' \in \mathscr{G}^m \setminus \{g\}$, a test arc $(s_{g'},t_g)$ is added to bind the parameters of the logical and integration functions, with net variable μ capturing as above the identity of module m:
    - if $(g',g) \in \mathscr{E}^m$ and $\sigma_{g',g} \notin \mathit{args}(\mathscr{K}_g)$, this arc is labelled by $\{(\mu,x_{g'_\mu})\}$,
    - if $(g',g) \in \mathscr{E}^m$ and $\sigma_{g',g} \in \mathit{args}(\mathscr{K}_g)$, this arc is labelled by $\{(\mu,x_{g'_\mu})\} \cup \{(\eta,x_{g'_\eta}) \mid \forall\eta : \mathbb{T}(\mu,\eta) > 0\}$,
    - if $(g',g) \notin \mathscr{E}^m$ and $\sigma_{g',g} \in \mathit{args}(\mathscr{K}_g)$, this arc is labelled by $\{(\eta,x_{g'_\eta}) \mid \forall\eta : \mathbb{T}(\mu,\eta) > 0\}$.

For the collection in Figure 7, we obtain the Petri net depicted in Figure 9.

4.3 Implementation

A prototype of the construction presented above has been implemented on the top of SNAKES toolkit [32]. SNAKES is a full featured Petri net library intended for quick prototyping; it uses Python programming language [33] to express the various Petri net annotations. Using this implementation:

- LRMs can be fully specified as Python classes;



$$M_0(s_A) \stackrel{\text{df}}{=} \{(0,1),(1,0),(2,1)\}$$
$$M_0(s_B) \stackrel{\text{df}}{=} \{(0,1),(1,1),(2,0)\}$$
$$M_0(s_C) \stackrel{\text{df}}{=} \{(0,0),(1,1),(2,0)\}$$

Fig. 9 The high-level Petri net representation of the CLRM of Figure 7. A possible initial marking may be M_0 as given below the Petri net.

- arbitrary integration functions can be user-defined, some are predefined;
- CLRMs can be defined by composing LRM instances with arbitrary topologies;
- CLRMs topology can be drawn with automatic layout;
- all possible stable states of a CLRM can be computed;
- reachable states from an initial one can also be computed, with extraction of the reachable stable states.

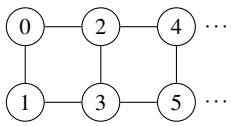
5 Modelling of interconnected LRMs using high-level Petri nets

In this section, we illustrate the modelling of interconnected logical regulatory modules by means of high-level Petri nets. In a first step, we analyse nine cases of collections of several copies of our toy LRM (as defined in Figure 5). We show how variations of the topological or the integration functions can affect the behaviours of the whole model. Then, we discuss the application of our framework to the modelling of the segment-polarity module involved in the segmentation of the *Drosophila* embryo.

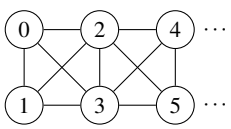
In both cases, the considered collection of modules is composed of copies of the same LRM. This indeed will be the case when modelling patches of identical cells, but one should be aware that the proposed framework does not impose such a restriction.

5.1 Interconnecting occurrences of the toy LRM

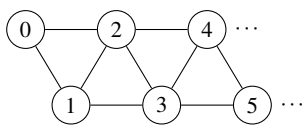
Let consider the toy LRM $\mathscr{M}$ as defined in Figure 5. In this section, we analyse the behaviours of a series of collections encompassing a number of occurrences of $\mathscr{M}$. All the considered collections are based on a tape of modules of width 2, with various lengths and topologies as depicted in Figure 10. Each topological relation $\mathbb{T}_k$ is so called because modules may have neighbours in k directions. Moreover, we denote $Toy(n,k)$ the CLRM containing $2 \cdot n$ copies of $\mathscr{M}$ arranged using $\mathbb{T}_k$ on a tape, as illustrated in Figure 10.



$$\mathbb{T}_4(i,j) \stackrel{\text{df}}{=} \mathbb{T}_4(j,i) \stackrel{\text{df}}{=} \begin{cases} 1 & \text{if } j = i+2 \vee (i \bmod 2 = 0 \wedge j = i+1) \\ 0 & \text{otherwise} \end{cases}$$



$$\mathbb{T}_8(i,j) \stackrel{\text{df}}{=} \mathbb{T}_8(j,i) \stackrel{\text{df}}{=} \begin{cases} 1 & \text{if } j = i+1 \vee j = i+2 \\ & \qquad \vee (i \bmod 2 = 0 \wedge j = i+3) \\ 0 & \text{otherwise} \end{cases}$$



$$\mathbb{T}_6(i,j) \stackrel{\text{df}}{=} \mathbb{T}_6(j,i) \stackrel{\text{df}}{=} \begin{cases} 1 & \text{if } j = i+1 \vee j = i+2 \\ 0 & \text{otherwise} \end{cases}$$

Fig. 10 The various topologies considered for 2-width tapes. On the left, the graphical view showing the organisation of the connexions. Definitions on the right side rely on the assumption that modules on the top line of the tape are numbered with consecutive even numbers, while modules on the bottom line are numbered (following the same direction) with consecutive odd numbers. All topologies considered here are symmetrical, the subscript k indicates the maximal number of neighbours in the topology $\mathbb{T}_k$ in an extended grid (e.g. with the topological relation $\mathbb{T}_4$, any module has at most 4 neighbours).

All stable states of systems $Toy(n,k)$ are such that the levels of the components in each module are equal (components A, B and C all equal to 0 or all to 1). So, to depict such a module stable state, we shall print a black and white grid, where black (resp. white) positions correspond to modules whose components are all 1 (resp. all 0), these positions respecting the topological relations of the modules. To simplify the presentation, we have also considered initial states that can be represented in this way.

Figure 11 shows the stable states that can be reached from chosen initial states. We can observe variations in the number of reachable states (which correspond to the size of the marking graph) and of reachable stable states; moreover, the stable states themselves are different.

All these examples have been obtained using the integration function $\sigma_{B,C}$ defined as $(y_i, d_i)_{1 \leq i \leq k} \rightarrow round(max(y_i \cdot d_i \mid 1 \leq i \leq k))$ (i.e. $\sigma_{B,C}$ evaluates to the maximal value of B in neighbouring modules; that is to say $\sigma_{B,C}$ is 1 provided that at least one neighbouring module encompasses a component B which level is 1). We may consider instead a function that returns 1 if at least two components B in the neighbouring modules are equal to 1, and return 0 otherwise. In this case, the behaviours of the collections might be rather different; in particular, the initial states considered for $Toy(3,4)$ and $Toy(4,4)$ (Figure 11, fourth and seventh rows) turn to be stable.

5.2 The Drosophila segment-polarity module

Early development of the fruit fly embryo is an ideal, very well-studied system for developmental biologists (see [48, 19] for good introductions to this topic). The embryo is organised into a series of segments along its antero-posterior axis (from the head to the tail). These segments will give rise to adult structures (legs, halteres, wings, etc.). This organisation is initiated by maternal morphogens, which control few dozens of genes involved in the initial segmentation. These genes have been split into several classes. The first classes, called *gap*, *pair-rule* and *segment-polarity* modules, constitute a temporal hierarchichal genetic system. Segment-polarity genes are under the control of the pair-rule genes. Their patterns of expression define the anterior and posterior parts of the segments and they are responsible for the stabilisation of the borders between embryonic segments (see Figure 12). The segment-polarity module involves about twenty genes, cross-regulations and intercellular signallings. It has been the subject of a wealth of theoretical modelling studies, in both continuous (e.g. [47, 22]) and logical (e.g. [2, 12, 35]) frameworks.

In [35], a logical model is defined and analysed, based on an intracellular interaction network of a dozen of components, submitted to two external inputs (the Wingless (Wg) and Hedgehog (Hh) signallings). Six copies of this module have been interconnected in a stripe to allow the representation of the different gene expression domains flanking the parasegmental borders (*parasegments* correspond to portions of two adjacent segments, see Figure 12). The cells are numbered from 1 to 6, with cell 3 denoting the cell just anterior to the border, cell 4 the cell just posterior to the border. This stripe of six cells represents a full parasegment traversed by the future border at its middle (see Figure 14). Connexions between these cells are set up through Hh and Wg signals, and in the wild-type case, these intercellular interactions are restricted to neighbouring cells. This fully defines the topological and integration functions of our collection of six modules.

In [35], the whole logical regulatory graph encompassing the six interconnected modules has been translated into its standard Petri net representation. This allowed us, by using INA [21], to verify the reachability of expected sta-

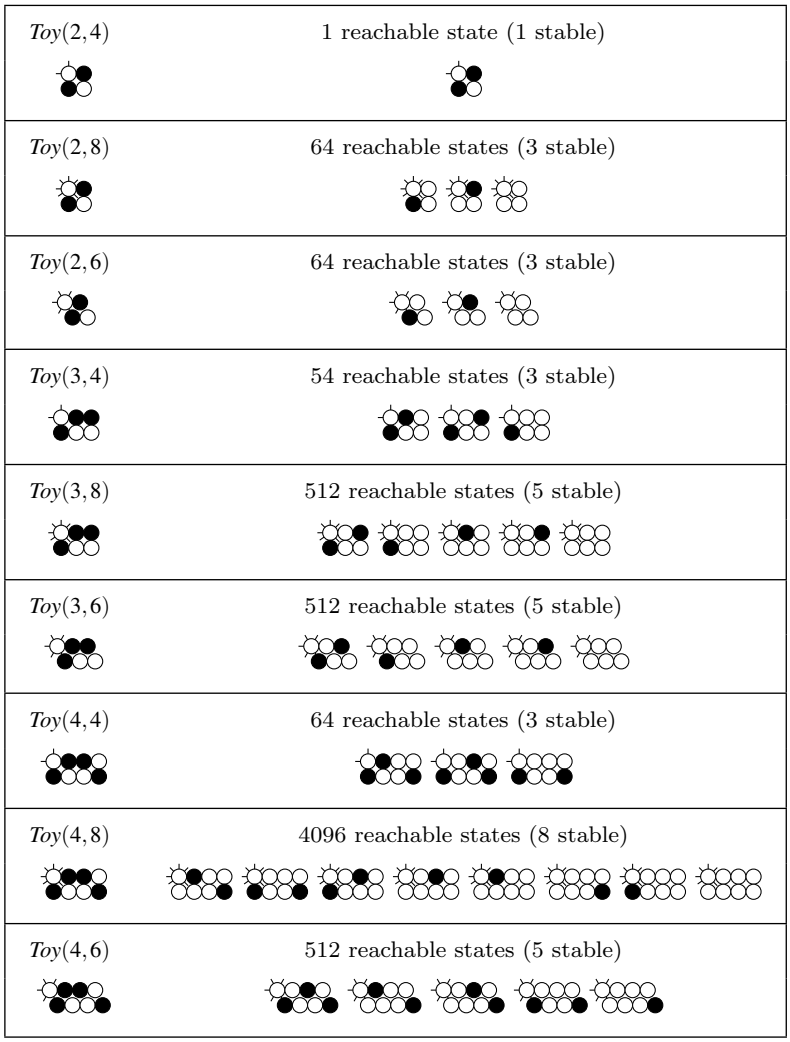


Fig. 11 Impact of the topological relation on the behaviours of the collections $Toy(n,k)$, defined on the LRM of Figure 5. In each row, a different combination of values of n and k (i.e. different number of modules and topological relation) is considered: on the left, the initial collection state is depicted, i.e. the initial state of each of its modules (black if internal A, B, C are all 1, white if they are all 0), and on the right, the total number of reachable states is given and the reachable stable states are depicted. The top-left module of each state is decorated with links showing the directions of its potential neighbours, e.g. in $\mathbb{T}(3,8)$, a module has possibly 8 neighbours (see Figure 10). This series of experiments shows that, for the same initial condition, a collection of interconnected modules can behave differently, depending on the signalling capacities (expressed here in terms of topological relations). For example, in the first collection $\mathbb{T}(2,4)$, the modules are not able to signal along the diagonal, contrary to the collection $\mathbb{T}(2,8)$. In the first, the initial pattern is stable, whereas it is not in $\mathbb{T}(2,8)$, which instead will stabilize in one of three other patterns. The integration function $\sigma_{B,C}$ considered here takes the maximal value of B over the neighbouring modules.

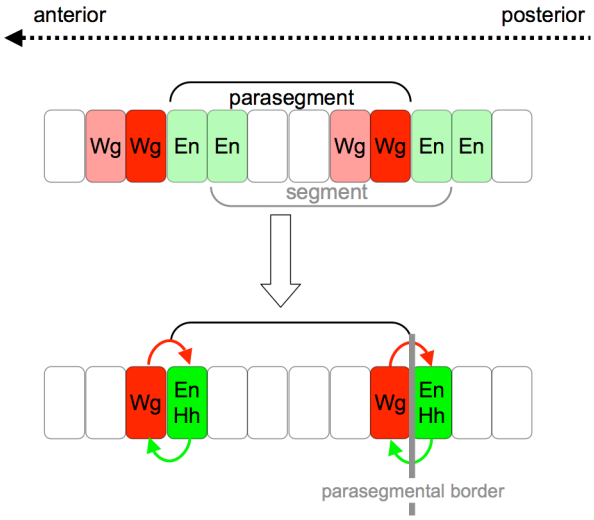


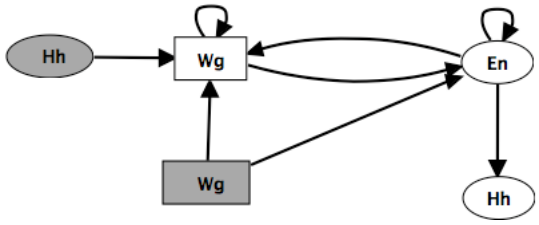
Fig. 12 Schematic illustration of the patterns of expression along the antero-posterior axis, in the last step of the genetic hierarchy controlling the *Drosophila* segmentation: (top) the antero-posterior axis; (middle) initial activation of En (Engrailed) and Wg (Wingless) by the pair-rule signals in stripes of cells (here, we considered that six cells are sufficient to represent the different regions of a parasegment, which includes the posterior part of a segment and the anterior part of the next segment; (bottom) the consolidation and refinement of the parasegmental border by the action of the polarity genes, which requires cell-cell communication.

ble states from a given initial pattern set up by the pair-rule module (see Figure 14). The proper construction of the model, by correctly interconnecting the six modules was rather cumbersome, justifying the development of a framework to support modular modelling.

In the context of this chapter, we will rely on a very much reduced version of the intracellular network. This reduced network has been obtained by the application of a systematic reduction method defined in [29], which preserves the essential dynamical properties (in particular the stable states). Our aim is not to give details on the properties of the segment-polarity module but rather to illustrate the potential of our framework. The LRM is depicted in Figure 13.

We study the reachable stable states for a collection of six cells arranged on a line as depicted in Figure 14. The initial state (defined by the action of the previous pair-rule module, as illustrated in Figure 12) is specified in Figure 14. We consider two topological relations (or signalling distances) and three integration functions, defined below.

- $\mathbb{T}_1$ defined such that $\mathbb{T}_1(i,j) \stackrel{\text{df}}{=} 1$ if $j = i+1$, $\mathbb{T}_1(i,j) \stackrel{\text{df}}{=} 0$ otherwise, and $\mathbb{T}(j,i) \stackrel{\text{df}}{=} \mathbb{T}(i,j)$ meaning that cells can only signal their immediate neighbours (plain lines in Figure 14);



Node	Target level	Logical rule
Wg	2	$((En=0 \wedge Wg=0 \wedge \sigma_{Wg,Wg}=2) \vee (En=0 \wedge Wg>1)) \wedge \sigma_{Hh,Wg}=1$
	0	otherwise
En	1	$((Wg=0 \wedge \sigma_{Wg,En}=2) \vee Wg>1) \wedge En=1$
	0	otherwise
Hh	1	$En=1$
	0	otherwise

Fig. 13 A reduced version of the segment-polarity logical regulatory module (obtained from the model of 12 components of Sánchez et al. [35], applying the reduction method presented in [29]). In this network, we kept the components exerting external signals (Wg and Hh) and the read-out genes of the action of this regulatory network (En and Wg). The external signals (coming from neighbouring cells) are denoted by gray nodes, rectangular nodes denote multi-valued components (Wg) and ellipse nodes denote Boolean components. The logical functions of the three internal components are given as logical rules.

- $\mathbb{T}_2$ defined such that $\mathbb{T}_2(i,j) \stackrel{\text{df}}{=} 1$ if $j=i+1$, $\mathbb{T}_1(i,j) \stackrel{\text{df}}{=} 1/2$ if $j=i+2$, $\mathbb{T}_j(i,j) \stackrel{\text{df}}{=} 0$ otherwise, and $\mathbb{T}(j,i) \stackrel{\text{df}}{=} \mathbb{T}(i,j)$, meaning that cells can signal their immediate neighbours (plain lines), but also their second neighbours, but with a lower weight (dotted lines in Figure 14).

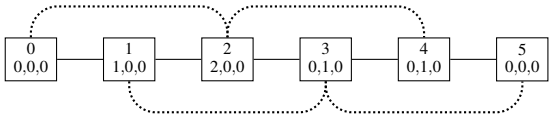


Fig. 14 A stripe of six cells with three cells accounting for the posterior region of a parasegment (cells 1, 2, and 3) and three cells accounting for the anterior region of the next parasegment (cells 4, 5, and 6), the parasegmental border being established between cells 3 and 4. In each cell, the initial levels of the components are indicated; each cellular state is given as a triple x_{Wg}, x_{En}, x_{Hh}. Plain lines denote where $\mathbb{T}_1$ and $\mathbb{T}_2$ are equal to 1 and dotted lines denote where $\mathbb{T}_2$ is equal to 1/2 (in other words, these lines denote the signalling capacities between the cells).

The integration functions we consider are:

- maximum weighted level:
$\sigma_{max} \stackrel{\text{df}}{=} (v_i, d_i)_{1 \le i \le k} \to round(max(v_i \cdot d_i \mid 1 \le i \le k))$

- maximum level of direct neighbours:
$\sigma_{max \ge 1} \stackrel{\text{df}}{=} (v_i, d_i)_{1 \le i \le k} \to max(v_i \mid 1 \le i \le k, d_i = 1)$

- maximum level for neighbours with weight at least 1/2:
$\sigma_{max \ge 1/2} \stackrel{\text{df}}{=} (v_i, d_i)_{1 \le i \le k} \to max(v_i \mid 1 \le i \le k, d_i \ge 1/2)$

For example, let consider the topology $\mathbb{T}_2$ and $\sigma_{Hh,Wg}$, which integrates the Hh signal acting on Wg. If $\sigma_{Hh,Wg} = \sigma_{max}$, the value of the integrated signal is the maximal level of Hh in neighbouring cells (including second neighbours). If $\sigma_{Hh,Wg} = sigma_{max \ge 1}$, the value of the integrated signal is the maximal level of Hh in direct neighbouring cells.

Figure 15 shows the results for various combinations of these parameters. When $\mathbb{T}_1$ is used, all neighbour signals have a weight 1, so we have $\sigma_{max} = \sigma_{max \ge 1} = \sigma_{max \ge 2}$.

In [35], the model considered for the six cell stripe was based on the assumption of short range Wg and Hh signals. Hence, to analyse the case of nkd loss-of-function (naked cuticle is one of the segment-polarity genes), it was necessary to consider a rewired network, accounting for the increased diffusion of Wg. In the framework presented here, such a change would only consist in modifying the topological function.

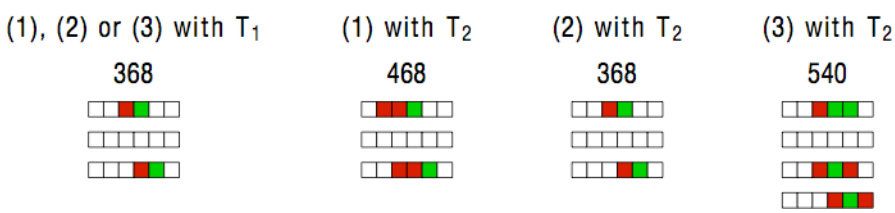


Fig. 15 Reachable states and reachable stable states for various configurations: (1) $\sigma_{Hh,Wg} = \sigma_{Wg,Wg} = \sigma_{Wg,En} = \sigma_{max}$; (2) $\sigma_{Hh,Wg} = \sigma_{Wg,Wg} = \sigma_{Wg,En} = \sigma_{max \ge 1}$; (3) $\sigma_{Hh,Wg} = \sigma_{max \ge 1}$ and $\sigma_{Wg,Wg} = \sigma_{Wg,En} = \sigma_{max \ge 1/2}$. For each configuration, we have indicated the total number of reachable state and drawn the reachable stable states. White squares correspond to cells where all the component levels are zero; red squares correspond to cells where $x_{Wg} = 2$ and $x_{Hh} = x_{En} = 0$; and green squares correspond to cells where $x_{Wg} = 0$ and $x_{Hh} = x_{En} = 1$. Note that the expected wild-type pattern as shown in the lower part of Figure 12, with a Wg expressing cell (red) and an En expressing cell (green) flancking the border, is recovered with topology $\mathbb{T}_1$ and with $\mathbb{T}_2$ if choosing $\sigma_{max \ge 1}$.

6 Conclusions

In this chapter, we have presented a modelling framework combining the logical formalism with Petri nets (PNs). The standard Petri net representation of logical regulatory graphs, although not very legible, enables the use of existing PN analysis tools such as INA. In this respect, GINsim provides export facilities to generate files in the format expected by PN tools (e.g. INA [21]).

A challenging problem arises when considering regulated metabolic networks. PNs open the way to a qualitative integrated modelling of regulated metabolic pathways as proposed in [40], properly connecting PN models of the biochemical pathway and the regulatory control (this being modelled as a logical regulatory graph expressed in terms of a PN).

Developmental processes relate to cell differentiation and pattern formation. In particular, in this chapter, we have delineated a framework that allows the modelling of patches of communicating cells. To illustrate the potential of this framework, we have considered the segment-polarity module involved in the segmentation of the *Drosophila* embryo. In such processes, one has to consider connexions of several (intra)cellular regulatory networks. We propose to define these individual regulatory networks as logical regulatory modules (LRMs), identifying input signals they can receive from the outside. Then, a collection of such modules (a CLRM) can be defined, by setting up the number and type of LRMs, as well as the rules governing their interconnexions. From such a CLRM, a large logical regulatory network can be recovered. This procedure could be easily implemented in GINsim.

More importantly, we have defined a compact and legible high-level Petri net (HLPN) representation of a CLRM. Implementation of the construction has been provided. This HLPN provides a flexible framework, which allows to easily model different configurations of a patch of cells, the impact of the topology (e.g. the range of a signal).

It is worth noting here, that the proposed framework allows the consideration of other situations than communicating cells within a patch. For example, in large regulatory networks, modularity arises from physical delimitation or cellular compartements (e.g. nucleus, cytoplasm) or from functional role (e.g. core cell cycle control).

Recently, a modular logical modelling of the budding yeast cell cycle has been delineated in [17]. In this case, modules corresponding to regulatory networks with distinct functional roles in the cell cycle control, have common components. Hence, while composing these modules, one has to properly set up the logical rules of these common components. Based on the methodology proposed in [17], the framework presented in this chapter could be extended to this case of overlapping modules.

Although modularity is now recognized as an important feature for large biological networks, little formal work and tool development support combination or composition of regulatory networks (see [39] and references therein).

In [39], the authors address this question and distinguish between fusion, composition, aggregation and flattening as distinct processes for building larger models from smaller ones. The HLPN framework presented in Section 4 can be viewed as a model aggregation, in that it is a reversible process (the modules are conserved). In terms of modelling tools, it is worth noting that ProMoT, a tool that eases the definition and edition of modular models, supports the logical formalism [27]. It is based on principles that are quite similar to those developed in Section 4.

The main motivation to develop the framework proposed here is to study how existing cellular processes are controlled. However, it could be useful in the field of synthetic biology that aims at designing novel artificial biological systems (see [15] and references therein). Indeed, synthetic biology relies on the concept of modularity, by conceiving building blocks and combining them [1]. So far, synthetic biology mainly designed intracellular gene networks, but synthetic multicellular systems involving cell-cell communication emerged in recent years (see e.g. [3]).

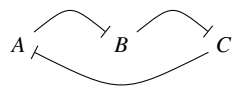
The complexity of regulatory networks dealt by modellers calls for the development of original and efficient computational means. Here, we have defined a framework that greatly facilitates the definition of models encompassing interconnected regulatory modules. However, we still need to make progresses to analyse such large models. Combining the logical and PN formalisms, as well as taking advantage of the modular structure of the models should allow the development of more efficient tools.

Acknowledgements We thank G. Batt, A. Naldi, E. Remy, S. Soliman, D. Thieffry for fruitful discussions. This work was supported by the French Research Agency (project ANR-08-SYSC-003).

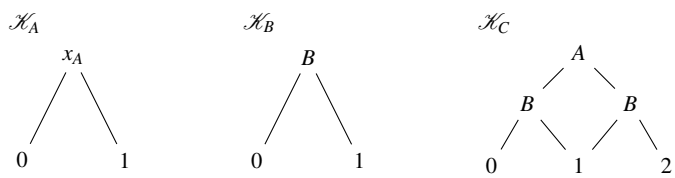
Problems

1. Give the MDD representation of the function f_2 given in Figure 2 considering the variables A and B in the reverse order (i.e. first B, then A). Then transform the obtained MDD into an IDD (labelling the edges with integer intervals, and merging them when possible as explain in Section 2.2.

2. Consider the genetic regulatory network called *repressilator* as defined by Elowitz and Leibler [16], which consists of three genes (denoted here A, B and C), connected in a negative circuit (each component represses its successor in the circuit, and is repressed by its predecessor): Assuming Boolean levels for A, B and C, all interaction thresholds are equal to 1, and the components share the same logical rule: $\mathscr{K}(0) = 1$ and $\mathscr{K}(1) = 0$, i.e. if the inhibitor is at level 0 the regulated component target level is 1, and the other way around. Draw the full STG of this model and verify that it encompasses a unique complex attractor, reachable from any initial state.



3. Consider an LRG encompassing a component C regulated by A and B, both positively auto-regulated, with $Max_A = Max_B = 2$ and $Max_C = 1$. The logical functions $\mathscr{K}_A$, $\mathscr{K}_B$ and $\mathscr{K}_C$ given by their MDD representations below (note that $\mathscr{K}_A$ and $\mathscr{K}_B$ are the identity function). Give the logical expressions defining these three functions (using the connectors $\wedge$ and $\vee$ as in Figure 2. Give the P/T representation of this LRG (verify that the autoregulations define transitions that are useless since they are never enabled, see [9]).

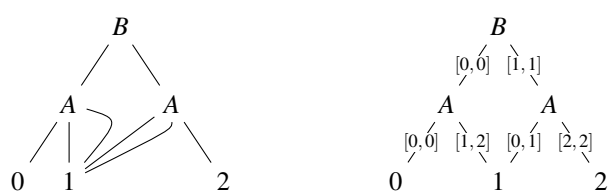


4. Find another enabling binding for the high-level Petri net depicted in Figure 8. Which module is involved by this binding and what are the corresponding levels x_A and x_B? Assume that $\delta_A(x_A) = 1$ and fire the transition accordingly.

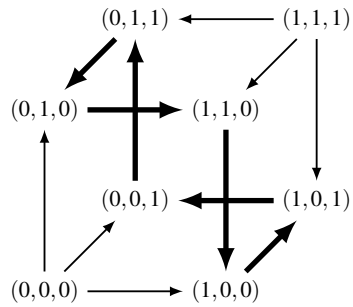
5. Construct the high-level Petri net representation of the CLRM depicted in Figure 14, including the initial marking. Indications:

1. Start with determining the arguments of each regulatory function, separating internal and external regulators.
2. Draw each Petri net transition separately, together with the places for the needed regulatory components.
3. Merge these Petri net parts by collapsing the places that implement the same regulators.
4. Add the initial marking, corresponding to the expression levels indicated in Figure 14.

Solution 1. 1 The MDD representation of f_2, considering the decision variables B and A in this order is given below (on the left side), with the corresponding IDD on the right.



Solution 2. The full STG contains 8 nodes (i.e. 2^3, since each of the 3 components can take 2 values) and is displayed below. The cyclical attractor is emphasised with bold arcs and contains 6 nodes. The only states which are not in the attractor are $(0,0,0)$ and $(1,1,1)$. These states are not reachable, and their succesors are all in the attractor. Hence, whatever the initial state, in at most one step, the system will be trapped in a cyclical trajectory, which consists in oscillations of the components A, B and C.

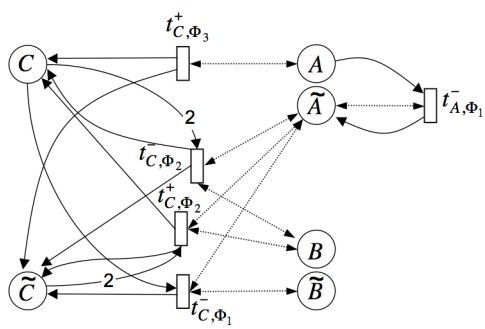


Solution 3. First, we give below the logical expressions of the functions:

$$\mathscr{K}_A(A) = \begin{cases} 0 \; if \; (A=0) \\ 1 \; if \; ((A=1) \end{cases} \qquad \mathscr{K}_B(B) = \begin{cases} 0 \; if \; (B=0) \\ 1 \; if \; ((B=1) \end{cases}$$

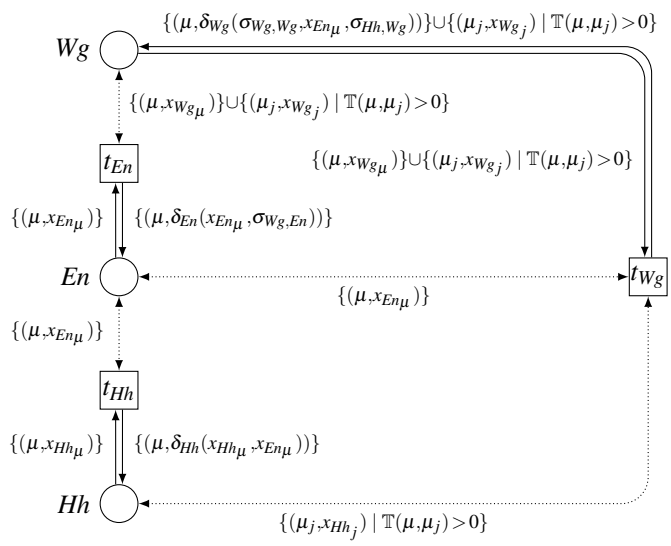
$$\mathscr{K}_C(A,B) = \begin{cases} 0 \; if \; (A=0) \wedge (B=0) \\ 1 \; if \; ((A=0) \wedge (B=1)) \vee ((A=1 \wedge (B=0)) \\ 2 \; if \; (A=1) \wedge (B=1) \end{cases}$$

The corresponding P/T net is given below. Note that transition t^-_{A,Φ_1}, which corresponds to the leftmost path in the MDD representation of $\mathscr{K}_A$ is never enabled (it requires that $x_A = 0$ i.e. one token must be present in place $\tilde{A}$, which makes impossible the presence of a token in A).



Solution 4. The alternative enabling binding is $\beta_2 = (\mu \to 1, x_{A_\mu} \to 0, x_{B_\mu} \to 0)$. This corresponds to $x_A = x_B = 0$ for module 1. The firing of transition t_A with β_2 leads to marking M' such that $M'(A) = \{(0,1),(1,1)\}$ and $M'(B) = \{(0,0),(1,0)\}$.

Solution 5. The high level Petri net representation of the segment polarity module interconnected as specified in Figure 14 is given below:



The initial marking is M defined for each place as:

$$M(Wg) \stackrel{\text{df}}{=} \{(0,0),(1,1),(2,2),(3,0),(4,0),(5,0)\}$$
$$M(En) \stackrel{\text{df}}{=} \{(0,0),(1,0),(2,0),(3,1),(4,1),(5,0)\}$$
$$M(Hh) \stackrel{\text{df}}{=} \{(0,0),(1,0),(2,0),(3,0),(4,0),(5,0)\}$$

References

1. C. Agapakis and P. Silver. Synthetic biology: exploring and exploiting genetic modularity through the design of novel biological networks. *Mol Biosyst*, 5(7):704–13, 2009.
2. R. Albert and H. G. Othmer. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster. *J. Theor. Biol.*, 223(1):1–18, 2003.
3. S. Basu, Y. Gerchman, C H. Collins, F H. Arnold, and R. Weiss. A synthetic multicellular system for programmed pattern formation. *Nature*, 434(7037):1130–4, 2005.
4. F. Bause, P. Kemper, and P. Kritzinger. Abstract Petri Net Notation. Technical report, Forschungsbericht Nr. 563 des Fachbereichs Informatik der Universität

Dortmund (Germany). Also in: Petri Net Newsletter No. 49, pages 9-27. October 1995., 1994.
5. R.E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Trans. Comput.*, 35(8):677–91, 1986.
6. C. Chaouiya, E. Remy, P. Ruet, and D. Thieffry. Qualitative modelling of genetic networks: From logical regulatory graphs to standard Petri nets. In *ICATPN'04*, volume 3099 of *Lecture Notes in Computer Science (LNCS)*, pages 137–56, 2004.
7. C. Chaouiya, E. Remy, and D. Thieffry. Qualitative Petri net modelling of genetic networks. *Trans Comp Syst Biol*, VI(4220):95–112, 2006.
8. CDSSZ-MC. Tools for the symbolic analysis of bounded petri nets, website. http://www-dssz.informatik.tu-cottbus.de/index.html?/software/mc.html.
9. C. Chaouiya, A. Naldi, E. Remy, and D. Thieffry. Petri net representation of multi-valued logical regulatory networks. *Natural Computing*, in revision.
10. C. Chaouiya, E. Remy, B. Mossé, and D. Thieffry. Qualitative analysis of regulatory graphs: a computational tool based on a discrete formal framework. In *POSTA'03*, volume 294 of *Lecture Notes in Control and Information Sciences (LNCIS)*, pages 119–26, 2003.
11. C. Chaouiya, E. Remy, and D. Thieffry. Petri net modelling of biological regulatory networks. *Journal of Discrete Algorithms*, 6(2):165–77, 2008.
12. M. Chaves, R. Albert, and E. D. Sontag. Robustness and fragility of boolean models for genetic regulatory networks. *J. Theor. Biol.*, 235(3):431–49, 2005.
13. JP. Comet, H. Klaudel, and S. Liauzu. Modeling multi-valued genetic regulatory networks using high-level Petri nets. In *ICATPN'05*, volume 3536 of *Lecture Notes in Computer Science (LNCS)*, pages 208–27. Springer-Verlag, 2005.
14. H. de Jong. Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput Biol.*, 1(9):67–103, 2002.
15. D A. Drubin, J C. Way, and P A. Silver. Designing biological systems. *Genes Dev*, 21(3):242–54, 2007.
16. M B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–8, 2000.
17. A. Fauré, A. Naldi, C. Chaouiya, A. Ciliberto, and D. Thieffry. Modular logical modelling of the budding yeast cell cycle. *Molecular BioSystems*, 5:1787–96, 2009.
18. A. Fauré, A. Naldi, C. Chaouiya, and D. Thieffry. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22(14):124–31, 2006.
19. S F. Gilbert. *Developmental Biology*. Sunderland, 8th edition, 2006.
20. GINsim. Gene interaction network simulation, sofware website. http://gin.univ-mrs.fr/GINsim.
21. INA. Integrated net analyzer, website. http://www2.informatik.hu-berlin.de/ starke/ina.html.
22. N. T. Ingolia. Topology and robustness in the drosophila segment polarity network. *PLoS Biol.*, 2(6):e123, 2004.
23. T. Kam, T. Villa, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. Multi-valued Decision Diagrams: Theory and applications. *International Journal on Multiple-Valued Logic*, 4:9–62, 1998.
24. S. Kauffman. Metabolic stability and epigenesis in randomly constructed genetics nets. *J. Theor. Biol.*, 22:437–67., 1969.
25. H. Klaudel and F. Pommereau. M-nets: a survey. *Acta Informatica*, 45:537–64, 2008.
26. L. Mendoza. A network model for the control of the differentiation process in Th cells. *Biosystems*, 84(2):101–14, 2006.
27. S. Mirschel, K. Steinmetz, M. Rempel, M. Ginkel, and E. D. Gilles. ProMoT: Modularmodeling for systems biology. *Bioinformatics*, 25(5):687–9, 2009.

28. A. Naldi, D. Berenguier, A. Fauré, F. Lopez, D. Thieffry, and C. Chaouiya. Logical modelling of regulatory networks with GINsim 2.3. *Biosystems*, 97(2):134–9, 2009.
29. A. Naldi, E. Remy, D. Thieffry, and C. Chaouiya. A reduction of logical regulatory graphs preserving essential dynamical properties. In *CMSB'09*, volume 5688 of *Lecture Notes in Bioinformatics (LNBI)*, pages 266–80, 2009.
30. A. Naldi, D. Thieffry, and C. Chaouiya. Decision diagrams for the representation of logical models of regulatory networks. In *CMSB'07*, volume 4695 of *Lecture Notes in Bioinformatics (LNBI)*, pages 233–47, 2007.
31. PNML.org. The reference site for the Petri Net Markup Language. http://www.pnml.org/.
32. Franck Pommereau. Quickly prototyping Petri nets tools with SNAKES. In *PNTAP'08*, ACM Digital Library, pages 1–10. ACM, 2008.
33. Python Software Foundation. Python programming language. http://www.python.org.
34. J. Saez-Rodriguez, L. Simeoni, JA. Lindquist, A. Hemenway, U. Bommhardt, B. Arndt, U-U. Haus, R. Weismantel, E. Gilles, S. Klamt, and B. Schraven. A logical model provides insights into T cell receptor signaling. *PLoS Comput Biol*, 3(8):e163, 2007.
35. L. Sánchez, C. Chaouiya, and D. Thieffry. Segmenting the fly embryo: a logical analysis of the segment polarity cross-regulatory module. *Int. J. Dev. Biol.*, 52(8):1059–75, 2008.
36. L. Sánchez and D. Thieffry. A logical analysis of the Drosophila gap-gene system. *J. Theor. Biol.*, 211(2):115–41, 2001.
37. L. Sánchez and D. Thieffry. Segmenting the fly embryo: a logical analysis of the pair-rule cross-regulatory module. *J. Theor. Biol.*, 224(4):517–37, 2003.
38. T. Schlitt and A. Brazma. Current approaches to gene regulatory network modelling. *BMC Bioinformatics*, 8(Suppl 6):S9, 2007.
39. CA. Shaffer, R. Randhawa, and JJ. Tyson. The role of composition and aggregation in modeling macromolecular regulatory networks. In *Proc of the Winter Simulation Conf*, pages 1628–36, 2006.
40. E. Simão, E. Remy, D. Thieffry, and C. Chaouiya. Qualitative modelling of regulated metabolic pathways: application to the tryptophan biosynthesis in E. Coli. *Bioinformatics*, 21(21 Suppl 2):190–6, 2005.
41. L.J. Steggles, R. Banks, O. Shaw, and A. Wipat. Qualitatively modelling and analysing genetic regulatory networks: a Petri net approach. *Bioinformatics*, 23(3):336–43, 2007.
42. K Strehl and L Thiele. Interval diagrams for efficient symbolic verification of process networks. *IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems*, 19(8):939–56, 2000.
43. R. Thomas. Boolean formalisation of genetic control circuits. *J. Theor. Biol.*, 42:565–83, 1973.
44. R. Thomas. Regulatory networks seen as asynchronous automata: A logical description. *J. Theor. Biol.*, 153:1–23, 1991.
45. R. Thomas, D. Thieffry, and M. Kaufman. Dynamical behaviour of biological regulatory networks–I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bull. Math. Biol.*, 57:247–76, 1995.
46. TINA. Time petri net analyzer, website. http://www.laas.fr/tina/.
47. G. von Dassow and G. M. Odell. Design and constraints of the drosophila segment polarity module: robust spatial patterning emerges from intertwined cell state switches. *J. Exp. Zool.*, 294(3):179–215, 2002.
48. L Wolpert, R Beddington, J Brockes, T Jessell, P Lawrence, and E Meyerowitz. *Principles of development.* Oxford University Press, 3rd edition, 2006.

Index

Binary Decision Diagrams, 5

Collection of interconnected Logical Regulatory Module, 15
Complex attractors, 9

Interval Decision Diagrams, 8

Logical formalism, 3
Logical Regulatory Graph, 3
Logical Regulatory Module, 13

Multi-valued Decision Diagrams, 6

Reduced Ordered Multi-valued Decision Diagrams, 5
Regulatory Networks, 3
Repressilator, 27

Segment-polarity module, 21
Stable states, 9
State Transition Graph, 8