Nat Comput (2011) 10:727–750
DOI 10.1007/s11047-010-9178-0

Petri net representation of multi-valued logical regulatory graphs

C. Chaouiya · A. Naldi · E. Remy · D. Thieffry

Published online: 21 February 2010

Abstract Relying on a convenient logical representation of regulatory networks, we propose a generic method to qualitatively model regulatory interactions in the standard elementary and coloured Petri net frameworks. Logical functions governing the behaviours of the components of logical regulatory graphs are efficiently represented by Multivalued Decision Diagrams, which are also at the basis of the translation of logical models in terms of Petri nets. We further delineate a simple strategy to sort trajectories through the introduction of priority classes (in the logical framework) or priority functions (in the Petri net framework). We also focus on qualitative behaviours such as multistationarity or sustained oscillations, identified as specific structures in state transition graphs (for logical models) or in marking graphs (in Petri nets). Regulatory circuits are known to be at the origin of such properties. In this respect, we present a method that allows to determine the functionality contexts of regulatory circuits, i.e. constraints on external regulator states enabling the corresponding dynamical properties. Finally, this approach is illustrated through an application to the modelling of a regulatory network controlling T lymphocyte activation and differentiation.

Keywords Gene regulation · Biological networks · Regulatory circuits · Logical modelling · Petri nets · Signal transduction · Cell differentiation

C. Chaouiya
Instituto Gulbenkian de Ciência, Oeiras, Portugal
e-mail: chaouiya@igc.gulbenkian.pt

C. Chaouiya (✉) · A. Naldi · D. Thieffry
INSERM U928—TAGC, Marseille, France

A. Naldi · D. Thieffry
Université de la Méditerranée, Marseille, France

E. Remy
Institut de Mathématiques de Luminy, Marseille, France

D. Thieffry
CONTRAINTES Project, INRIA-Paris-Rocquencourt, Le Chesnay, France

1 Introduction

Most essential cellular processes are controlled by regulatory networks involving diverse regulatory interactions, e.g. transcriptional regulations of target genes, protein modifications, diffusion and sequestring of signalling molecules. Due to the growing complexity of these networks, proper understanding of their dynamical properties requires the development of adequate formal representations, as well as efficient modelling and analysis tools. Approaches generally used to model such regulatory networks include graph theory, Boolean networks, differential equations, or yet stochastic equations (see reviews in de Jong 2002; Schlitt and Brazma 2007). It is worth noting that current data on the molecular processes governing regulatory interactions remains largely qualitative. Indeed, precise information on kinetic parameters or concentration levels are often lacking. This is particularly true for the regulation of gene expression, as the role of a regulatory product is often just characterised as activating or inhibiting a target gene in a given context. The semantics associated with biological interactions varies: while in a chemical reaction, the reactants are consumed, the expression levels of a transcriptional regulator seldomly change during the regulatory process. One successful approach to qualitatively model such regulatory networks is the *logical method* initially developed by Thomas and co-workers (Thomas and D'Ari 1990; Thomas 1991; Thomas et al. 1995; Chaouiya et al. 2003) (see also the related approach in Kauffman 1993). Indeed this method was productively used to model a diversity of regulatory interactions, beyond transcriptional genetic regulation, as demonstrated in Fauré et al. (2006, 2009), González et al. (2008), and Sánchez et al. (2008). However, we face a classical combinatorial explosion when we analyse large networks. This is one of the motivations that drove us to propose a systematic translation of logical regulatory models into Petri nets (PNs), to take advantage of the corresponding simulation and analysis tools. Another interesting prospect of the PN representation of regulatory networks lies in the coupling of metabolic pathways with regulatory processes acting upon these pathways (see Simão et al. 2005) for a first step in this direction). Finally, PNs open the way to quantitative extensions, e.g. adding deterministic or stochastic rates on the transitions, assigning non-negative real values to places (Goss and Peccoud 1998; Srivastava et al. 2001; Mura and Csikasz-Nagy 2008; Heiner et al. 2008), or considering hybrid models (Nagasaki et al. 2004; Doi et al. 2006).

So far, PNs have been mainly employed to model and analyse metabolic networks, since PNs are well adapted to the representation of the consumption/production semantics related to chemical reactions. The representation of a regulatory relation by means of standard PNs is less straighforward. Section 2 introduces the logical approach for the modelling of regulatory networks. This allows us to define the representation of logical regulatory graphs by means of standard or coloured PNs (Sect. 3). One strategy to ease the analysis of large regulatory networks consists in cutting irrelevant trajectories thanks to the consideration of priority classes (Fauré et al. 2006). This drives us to discuss the introduction of such priorities in our models of regulatory networks in Sect. 4. Another interesting point of the logical framework is the establishment of the relationships between the occurrence of regulatory circuits and specific dynamical properties (see Thieffry 2007 and references therein). In Sect. 5, we recall how regulatory structures relate to multistationarity and oscillatory behaviours. The determination of functionality contexts is explained and its transposition in the context of PN representation of logical regulatory graphs is discussed. All these concepts are illustrated by a biological application dealing with a simplified model of the control of T lymphocyte activation and differentiation. The

paper ends with a discussion of future prospects regarding the use of Petri nets for the modelling and analysis of complex biological regulatory networks.

2 Multi-valued logical modelling of regulatory networks

Regulation refers to the molecular mechanisms responsible for changes in concentration or activity of a functional product. Such mechanisms range from transcriptional regulation to protein modifications (Alberts et al. 2008). Regulation functions (or response functions) are usually represented by sigmoid or step functions (Thomas and D'Ari 1990). It is then assumed that the regulatory effect becomes noticeable when the level of the regulator reaches a given threshold. Hence it makes sense to consider regulatory networks as discrete event systems. In this context, we rely on the logical modelling of regulatory networks introduced by Thomas and co-workers (Thomas 1991; Thomas et al. 1995; Chaouiya et al. 2003). In this framework, a discrete variable is associated to each regulatory component to represent its qualitative levels of expression (for a gene) or of activity (for a protein), and logical rules define the behaviours of the regulatory components as functions of the levels of their regulators.

Definition 1 *A logical regulatory graph (LRG)* is a directed labelled multigraph[1] $\mathcal{R} = (\mathcal{G}, \mathcal{M}ax, \mathcal{E}, \Theta, \mathcal{K})$ where,

- $\mathcal{G} = \{g_1, \ldots, g_n\}$ is the set of nodes, representing *genes* (or, more generally, *regulatory components*).
- $\mathcal{M}ax : \mathcal{G} \to \mathbb{N}^*$ associates a maximum level[2] $\mathcal{M}ax(g_i) = Max_i$ to node g_i. In the sequel, x_i denotes the *current level* of g_i ($x_i \in \{0, \ldots, Max_i\}$), a state of the system is thus given as a vector $x = (x_i)_{i=1,\ldots,n}$.
- $\mathcal{E}$ is a finite multiset of ordered pairs of elements of $\mathcal{G}$ representing *regulatory interactions*. If $Max_i > 1$, g_i may have different effects onto a component g_j, depending on level x_i. Hence, the arc connecting g_i to g_j may be a multi-arc encompassing different interactions. The multiplicity of the arc (g_i, g_j) (i.e. the number of its constitutive interactions), is denoted $m_{i,j}$ ($1 \leq m_{i,j} \leq Max_i$). Loops (even multi-loops) are allowed: an arc (g_i, g_i) denotes a self-regulation of g_i.
- Θ is a labelling function, which associates a *threshold* to each element of $\mathcal{E}$. More precisely, $\theta_{i,j,k}$ is associated to the kth interaction between g_i and g_j (denoted (g_i, g_j, k), $k \in \{1, \ldots, m_{i,j}\}$), with $1 \leq \theta_{i,j,1} < \cdots < \theta_{i,j,m_{i,j}} = Max_i$. This interaction is *active*, when x_i, the level of its source g_i lays between the threshold of this interaction and that of the next interaction: $\theta_{i,j,k} \leq x_i < \theta_{i,j,k+1}$ (by convention, $\theta_{i,j,m_{i,j}} + 1 = Max_i + 1$). In other words, the interaction (g_i, g_j, k) is active in all states x for which $\theta_{i,j,k} \leq x_i < \theta_{i,j,k+1}$. For each $g_j \in \mathcal{G}$, $Reg(j)$ denotes the set of its regulators: $g_i \in Reg(j)$ if and only if $(g_i, g_j) \in \mathcal{E}$.
- $\mathcal{K} = (\mathcal{K}_1, \ldots, \mathcal{K}_n)$ defines the *logical rules* attached to the nodes specifying their behaviours: each $\mathcal{K}_i$ is a multi-valued logical function that gives the target value of g_i (the value to which g_i should tend), depending on the interactions acting on g_i at any state:

[1] Multigraphs are also called *pseudographs*.

[2] $\mathbb{N}^*$ is the set of non-zero natural numbers.

$$\mathcal{K}_i : \left(\prod_{g_j \in Reg(i)} \{0, \ldots, m_{j,i}\} \right) \rightarrow \{0, \ldots, Max_i\}.$$

Later on, when adequate, g_i will be denoted by i and $(i, j, \theta_{i,j,k})$ will stand for interaction (g_i, g_j, k). Also for convenience, since we work in a discrete framework, we will use the notation $[a, b]$ to represent the integer interval $\{a,\ldots, b\}$ $(a, b \in \mathbb{N}, a \leq b)$.

Note that the biologists often consider two types of interactions: activations have a positive effect on their targets, whereas repressions have a negative effect on their targets. However the actual effect of an interaction may depend on the presence of co-factors; its sign may even change depending on the context. In any case, the interactions and their signs can be derived from the logical functions.

We represent the behaviour of a LRG by a *state transition graph*, where nodes correspond to states (i.e. vectors encompassing the levels of the regulatory components), whereas arcs denote transitions between states. This graph is computed using the functions which indicate the transitions leading from the current state to its potential successor states (see Definitions 3 and 4). Here, we consider an asynchronous updating, where each transition corresponds to a change of a unique variable (see Chaouiya et al. 2003 for further details). This choice may lead to non-deterministic behaviours.

Definition 2 A state x of the LRG $\mathcal{R}$ is a n-tuple $(x_1,\ldots, x_n)$ of the levels of the regulatory components. The set of all potential states (or state space) is denoted by $\mathcal{S} = \prod_{i=1}^{n} [0, Max_i]$.

From Definition 1 it follows that a state $x \in \mathcal{S}$ fully determines, for each g_i, the set of active incoming interactions (hence there is no memory of past states). The next definition specifies the behaviour of each regulatory node as a function defined on the state space $\mathcal{S}$.

Definition 3 Given a LRG $\mathcal{R} = (\mathcal{G}, \mathcal{M}ax, \mathcal{E}, \Theta, \mathcal{K})$, the dynamics of $g_i \in \mathcal{G}$ is defined by:

$$\begin{aligned} \mathcal{F}_i^{\mathcal{K}} : \mathcal{S} &\longrightarrow [0, Max_i] \\ x &\mapsto \mathcal{K}_i \left(\left(\sum_{k=1}^{m_{j,i}} k \mathbf{1}_{[\theta_{j,i,k}, \theta_{j,i,k+1}[}(x_j) \right)_{j \in Reg(i)} \right), \end{aligned}$$

where $\mathbf{1}$ denotes the indicator function.

In the rest of the paper, to simplify the notations and because of the unique correspondance between $\mathcal{K}_i$ and $\mathcal{F}_i^{\mathcal{K}}$, we will refer to the dynamics as $\mathcal{K}_i$.

Remark 1 For all $g_i \in \mathcal{G}$, the dynamics $\mathcal{K}_i$ only depends on the values of x_j for $j \in Reg(i)$. In other words, $\mathcal{K}_i(x) = \mathcal{K}_i(x')$, for all $x, x' \in \mathcal{S}$ such that $\forall j \in Reg(i)$, $x_j = x'_j$.

Definition 4 Given a LRG $\mathcal{R} = (\mathcal{G}, \mathcal{M}ax, \mathcal{E}, \Theta, \mathcal{K})$ and a set of initial states $Init \subseteq \mathcal{S}$, the (asynchronous) state transition graph $(\mathcal{S}_{Init}, \mathcal{T})$ is the (finite) directed graph defined as follows:

- $\forall x \in Init$, $x \in \mathcal{S}_{Init}$,
- $\forall x \in \mathcal{S}_{Init}$, $\exists i | \mathcal{K}_i(x) \neq x_i \Rightarrow x' \in \mathcal{S}_{Init}$ s.t. $(x, x') \in \mathcal{T}$ and:

$$\begin{cases} x'_j = x_j & \forall j \neq i, \\ x'_i = x_i + 1 & \text{if } \mathcal{K}_i(x) > x_i, \\ x'_i = x_i - 1 & \text{if } \mathcal{K}_i(x) < x_i. \end{cases} \quad (1)$$

In other words, given a state x, the level of each g_i tends toward the *target value* given by $\mathcal{K}_i(x)$. If this is greater (*resp.* lower) than x_i (the current value of g_i), there is a call to increase (*resp.* decrease) by one the value of g_i, hence a transition towards a new state having the corresponding updated value for x_i.

It is worth noting that one might consider $Init = \mathcal{S}$ (as defined in Definition 2) and then obtain the *whole* state transition graph, encompassing all potential states and possible transitions defined by $\mathcal{K}$.

Terminal strongly connected components in the state transition graph denote regions of the state space where the system is eventually trapped, which we call *attractors*. These attractors might be stable states (states with no successor, corresponding to stable patterns of expressions in a gene regulatory network), or encompass states involved in elementary cycles or in intertwined cycles (these components indicate stable oscillations of the system, or yet homeostasis that ensures the maintenance of a certain equilibrium).

Throughout the paper, *state transition graphs* will refer to the dynamics of LRGs as defined above, whereas *marking graphs* will be used in the context of Petri net models.

For several years, we have been developing GINsim (Naldi et al. 2009a), a software dedicated to the definition and analysis of logical models. Performance considerations led us to propose the use of Reduced Ordered Multi-valued Decision Diagrams (ROMDDs, denoted MDDs from now on) to internally represent logical functions (Naldi et al. 2007). In Garg et al. (2007), decision diagrams were used to represent state transition graphs and analyse the dynamical properties of Boolean models. In our context, the MDD representation further facilitates the translation of logical models into Petri nets. In particular, it is used by GINsim modules exporting logical models into several PN formats.

To apply efficient algorithms for the analysis of our models, the function $\mathcal{K}_i$, which takes its values in $[0, Max_i]$, is represented in terms of a MDD with the levels x_j of the regulators $g_j \in Reg(i)$ as decision variables (Kam et al. 1998; Naldi et al. 2007) (see Fig. 1, bottom panel). It is possible to further compact this MDD by merging consecutive edges leading to the same child as explained hereafter.

Following the classical MDD representations, given $g_i \in \mathcal{G}$, for each decision variable x_j that appears in the diagram of $\mathcal{K}_i$, there are $Max_j + 1$ outgoing edges, implicitly labelled with the corresponding value in $[0, Max_j]$. Edges labelled with consecutive values pointing towards the same child can be merged 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. 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 $\mathcal{K}_i$ is v_Φ:

- If path Φ encompasses an edge going out the decision variable x_j, the set of assignments of x_j equals 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 simplification of the MDD), it means that $\mathcal{K}_i(x) = v_\Phi$ does not depend on x_j.

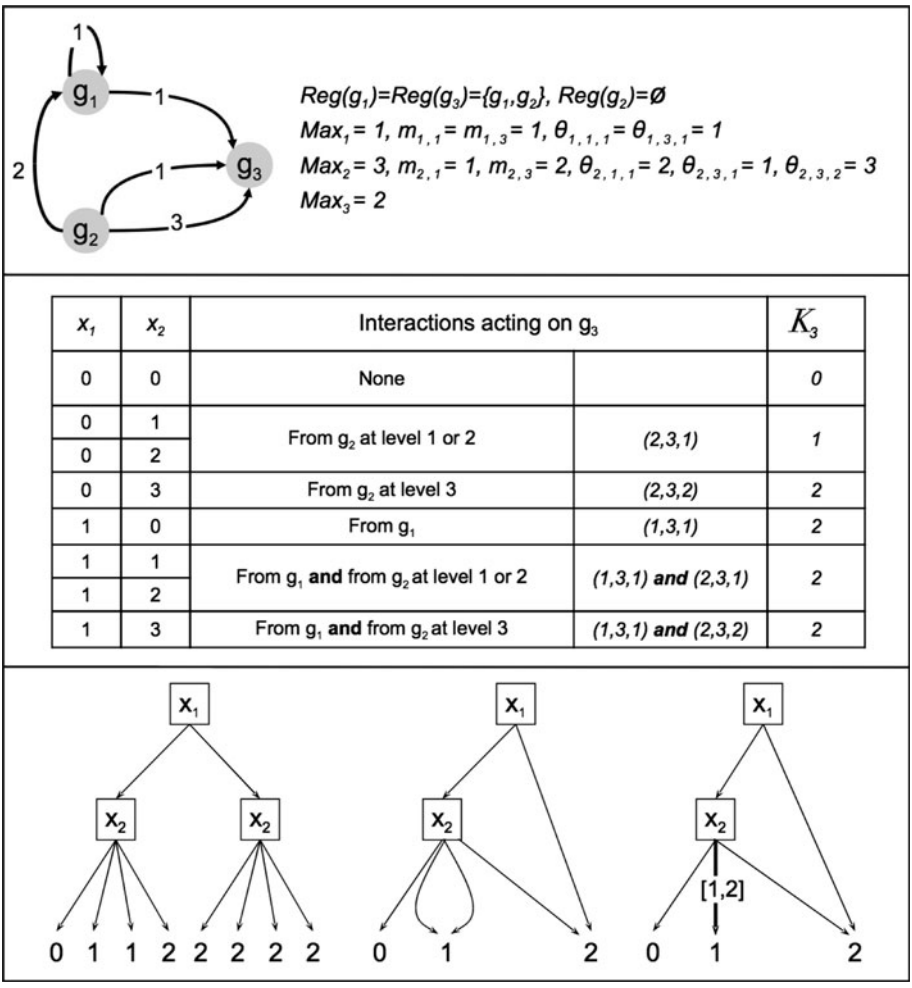


Fig. 1 Example of a *logical regulatory graph*. The *top panel* displays interactions between three nodes along with the specification of the thresholds and maximal levels. The *middle panel* shows the function $\mathcal{K}_3$, which defines the target value of g_3 depending on the levels of its regulators g_1 and g_2 (equivalently, depending on the combinations of active interactions). Definitions of $\mathcal{K}_1$ and $\mathcal{K}_2$ are omitted here. The *bottom panel* displays the decision tree representing $\mathcal{K}_3$, the corresponding reduced decision diagram (MDD), and, on the right, an illustration of the merging of consecutive edges linked to the same child

The use of MDD generally leads to a simplified expression of $\mathcal{K}_i$, but the resulting diagram and its complexity may vary depending on the ordering of the decision variables (see e.g. Kam et al. 1998; Fig. 5 for an illustration).

Figure 1 illustrates a logical graph, the function $\mathcal{K}_3$ of the node g_3, as well as the decision tree and resulting decision diagrams representing $\mathcal{K}_3$. Note that, hereafter, we sometimes depict MDDs as non-completely reduced decision diagrams to facilitate their interpretation.

Remark 2 Given a function $\mathcal{F} : \mathbb{N}^n \to \mathbb{N}^n$, one can recover a LRG with n regulatory components, the maximal values, the logical rules, the interactions and their thresholds.

The recovered interactions will all be *functional* interactions, i.e. interactions with an effect on the levels of their targets observed through the dynamics $\mathcal{F}$ (see Sect. 5 for the notion of functionality).

3 Petri net representation of logical models

Petri nets consumption/production semantics is well adapted to the representation of chemical reactions. However, it is less obvious to use the standard PN framework to represent regulatory mechanisms, where (1) there is no modification of the activity level of a regulator acting on its target, (2) the absence of a regulator may have an effect on its target. One could use inhibitory arcs to take into account this last situation, but we aim at proposing a representation using standard elementary Petri nets to take advantage of their powerful analysis framework. Because the level of each component in a LRG is bounded, we can use complementary places instead of inhibitory arcs. In a first step, a rewriting of LRGs into standard PNs is briefly presented. Next, we define a coloured PN version of this translation.

3.1 Standard Petri net representation

The definition below allows the explicit construction of a Petri net from a LRG, based on the MDD representation of the dynamics $\mathcal{K}_i$, where edges are labelled by integer intervals. The resulting Petri net has a behaviour equivalent to that of the original logical model (Property 1). Further details, basic properties and applications of this PN representation of LRGs are provided in Chaouiya et al. (2004) for the Boolean case, and in Chaouiya et al. (2006) for the multi-valued case. Note that the examples presented throughout this section are not meant to be realistic, but rather to illustrate the rules governing the rewriting and facilitate the understanding.

Definition 5 Given a LRG $\mathcal{R} = (\mathcal{G}, \mathcal{M}ax, \mathcal{E}, \Theta, \mathcal{K})$, we define the corresponding Multi-valued Regulatory Petri Net (MRPN) as follows:

- For each $g_i \in \mathcal{G}$, there are two places g_i and $\widetilde{g_i}$ that satisfy, for all markings M:

$$M(g_i) + M(\widetilde{g_i}) = Max_i, \tag{2}$$

meaning that $\widetilde{g_i}$ is the complementary place of g_i.

- For $g_i \in \mathcal{G}$, for each path Φ from the root to a leaf of the MDD representing $\mathcal{K}_i$, at most two transitions are defined, one accounting for the increasing tendency (denoted $t^+_{i,\Phi}$), the second accounting for the decreasing tendency (denoted $t^-_{i,\Phi}$). This simplifies when the leaf is associated with an extreme value (see below). For the considered component g_i, a path Φ defines assignment intervals of the levels of g_j in $Reg(i)$: $x_j \in [\phi_j, \phi'_j]$, where $\phi_j, \phi'_j \in [0, Max_j]$ 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 $Max_j - \phi'_j$.

When $\phi_j = \phi'_j$, from Eq. 2, it suffices to consider only one of these test arcs. If $[\phi_j, \phi'_j] = [0, Max_j]$, the decision variable should be omitted in the reduced MDD, because all possible values lead to the same result (hence places g_j and $\widetilde{g_j}$ are not connected to $t^+_{i,\Phi}$ nor to $t^-_{i,\Phi}$).

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 target 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 more than the target 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 $g_i \in \mathcal{G}$ and Φ a path in the decision diagram associated to $\mathcal{K}_i$, when $v_\Phi = 0$ or $v_\Phi = Max_i$ (the value of $\mathcal{K}_i$ 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. Figure 2 illustrates the standard PN representation of LRGs.

Remark 3 Regarding self-regulations, in Petri nets, all transitions are meant to represent events that effectively change the state of the modelled system. If g_i is a self-regulator, then it appears as a decision variable in the diagram of $\mathcal{K}_i$. If Φ is a path leading to a value v_Φ and if g_i is assigned to value v_Φ (i.e. $\phi_i = v_\Phi$), then no transition will be generated for Φ. Figure 3 illustrates this situation.

Property 1 *In the state transition graph $(\mathcal{S}, \mathcal{T})$ of a LRG $\mathcal{R} = (\mathcal{G}, \mathcal{M}ax, \mathcal{E}, \Theta, \mathcal{K})$, there exists a transition between two states x and x' iff there exists an enabled transition t in the associated MRPN defined as in Definition 5 such that $M[t\rangle M'$ (t is enabled by the marking M and its firing leads to the new marking M') with, for all $k = 1, \ldots n$:*

$$M(g_k) = x_k \quad M(\widetilde{g_k}) = Max_k - x_k,$$
$$M'(g_k) = x'_k \quad M'(\widetilde{g_k}) = Max_k - x'_k.$$

Proof Let consider $x, x' \in \mathcal{S}$ such that $(x, x') \in \mathcal{T}$, and let M be the marking of the associated MRPN such that $M(g_i) = x_i$, $M(\widetilde{g_i}) = Max_i - x_i$, for all $g_i \in \mathcal{G}$. We show first that this marking enables a transition that, when fired, leads to a new marking M' with $M'(g_i) = x'_i$, $M'(\widetilde{g_i}) = Max_i - x'_i$, for all $g_i \in \mathcal{G}$.

From Definition 4, there exists an i such that $x_i = x'_i \pm 1$ and $\mathcal{K}_i(x) \neq x_i$. The state x determines a unique decision path $\Phi(x)$ in the MDD representing $\mathcal{K}_i$. Recall that if the decision variable x_j has no interval assignment $[\phi_j(x), \phi'_j(x)]$ in the path $\Phi(x)$, places g_j and $\widetilde{g_j}$ are not connected to the transitions $t^+_{i,\Phi(x)}$ and $t^-_{i,\Phi(x)}$. Let $v_{\Phi(x)} = \mathcal{K}_i(x) \in [0, Max_i]$. Then,

1. If $0 \leq v_{\Phi(x)} < x_i$, then $M(g_i) \in [v_{\Phi(x)} + 1, Max_i]$. Moreover, for all $g_j \in Reg(i)$ such that the decision variable x_j is assigned in $\Phi(x)$, we have:

$$x_j = M(g_j) \in [\phi_j(x), \phi'_j(x)],$$
$$Max_j - x_j = M(\widetilde{g_j}) \in [Max_j - \phi'_j(x), Max_j - \phi_j(x)].$$

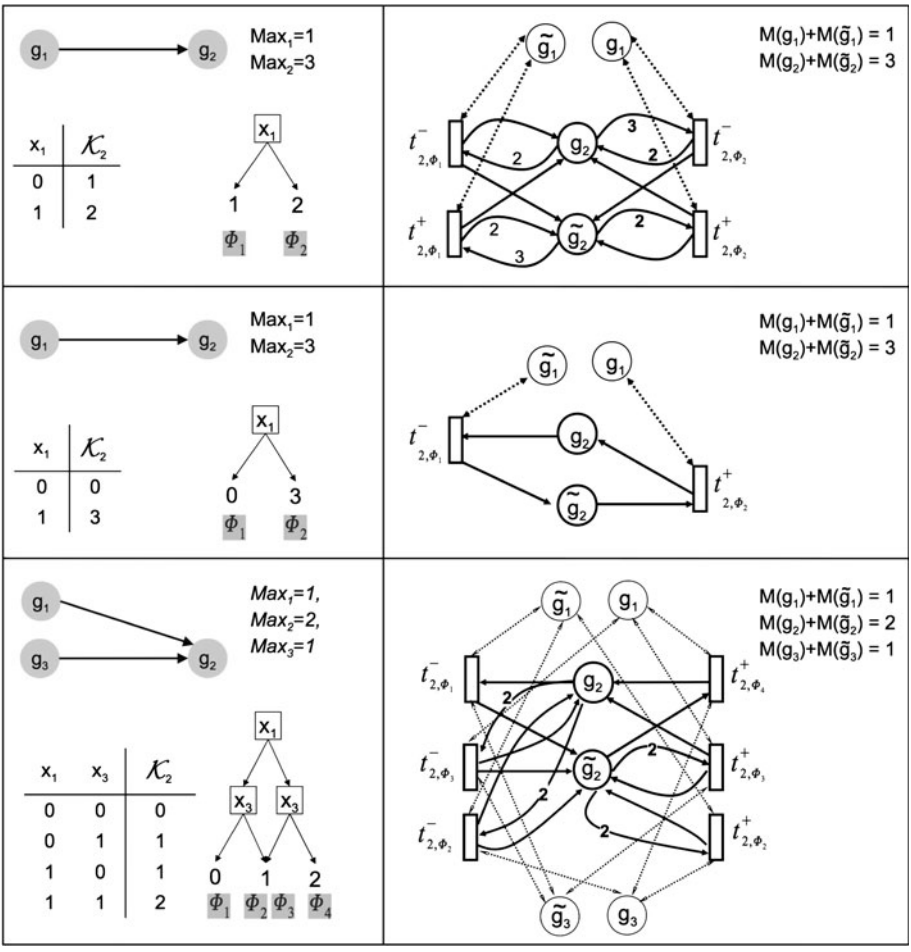


Fig. 2 LRG translations into standard PNs. The *top panel* displays a component g_1 (with 2 levels) that regulates component g_2 (4 levels); $\mathcal{K}_2$ is given in the form of a table and the corresponding MDD. When g_1 is present, g_2 tends to 2, while the *basal value* of g_2 is 1 (i.e. the value of g_2 in the absence of its regulator g_1). On the right side, the corresponding MRPN is displayed. For each component, two *complementary* places are defined. For each path in the decision diagram of $\mathcal{K}_2$, two transitions are defined: e.g., $\mathcal{K}_2(1) = 2$ is represented by t^+_{2,Φ_2} and t^-_{2,Φ_2}. If g_1 is marked (g_1 at level 1) and g_2 not marked (g_2 at level 0, 3 tokens in $\widetilde{g_2}$), then t^+_{2,Φ_2} is enabled and its firing *increases* the marking of g_2; if g_1 is marked and g_2 has 3 tokens (its highest level), then t^-_{2,Φ_2} is enabled and its firing *decreases* the marking of g_2. In the *middle panel* the same toy example is considered but with $\mathcal{K}_2$ taking extreme values. On the right side, the corresponding MRPN is simpler with only one transition for each situation (t^+_{2,Φ_1} and t^-_{2,Φ_2} are of no use here). The *bottom panel* gives a third example, with a resulting MRPN encompassing one transition for each path leading to extreme values (Φ_1 and Φ_4), and two transitions for each path leading to the intermediate value 1 (Φ_2 and Φ_3)

Hence $t^-_{i,\Phi(x)}$ is enabled (if $v_{\Phi(x)} \neq 0$, $\Phi(x)$ also defines a transition $t^+_{i,\Phi(x)}$ that is not enabled in M because $M(\widetilde{g}_i) \in [0, v_{\Phi(x)}]$).

2. If $x_i < v_{\Phi(x)} \leq Max_i$, then $M(\widetilde{g}_i)(= Max_i - x_i) \in [Max_i - v_{\Phi(x)} + 1, Max_i]$. Moreover, for all $g_j \in Reg(i)$ such that the decision variable x_j is assigned in $\Phi(x)$, we have:

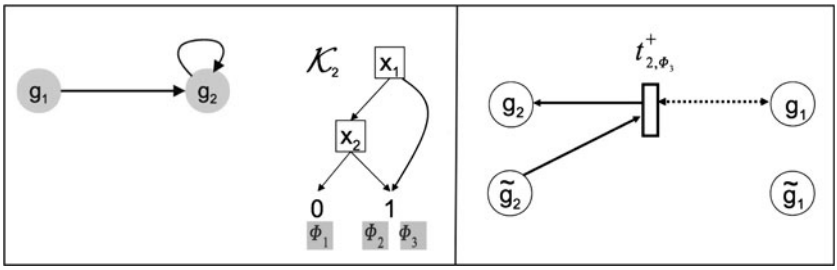


Fig. 3 Representation of a self-regulation in Petri nets. In the example shown, there is no need to define transitions for g_2 self-maintainance, i.e. transitions related to the paths: $\Phi_1 = (g_1 = 0,\ g_2 = 0)$ and $\Phi_2 = (g_1 = 0,\ g_2 = 1)$

$$x_j = M(g_j) \in [\phi_j(x), \phi'_j(x)],$$
$$Max_j - x_j = M(\widetilde{g_j}) \in [Max_j - \phi'_j(x), Max_j - \phi_j(x)].$$

Hence $t^+_{i,\Phi(x)}$ is enabled (if $v_{\Phi(x)} \neq Max_i$, $\Phi(x)$ also defines a transition $t^-{}_{i,\Phi(x)}$ that is not enabled in M because $M(g_i) \in [0,\ v_{\Phi(x)}]$).

3. When $t^-_{i,\Phi(x)}$ is fired ($v_{\Phi(x)} < x_i$),

$$M'(g_i) = M(g_i) - 1 = x_i - 1 = x'_i, \quad M'(\widetilde{g_i}) = M(\widetilde{g_i}) + 1 = Max_i - x_i + 1 = Max_i - x'_i.$$

4. When $t^+_{i,\Phi(x)}$ is fired ($x_i < v_{\Phi(x)}$),

$$M'(g_i) = M(g_i) + 1 = x_i + 1 = x'_i, \quad M'(\widetilde{g_i}) = M(\widetilde{g_i}) - 1 = Max_i - x_i - 1 = Max_i - x'_i.$$

Reciprocally, let consider a marking M enabling a transition t, and M' the marking such that $M[t > M'$, then there exists i such that:

$$M'(g_i) = M(g_i) + 1 \quad \text{and} \quad M'(\widetilde{g_i}) = M(\widetilde{g_i}) - 1,$$
$$\text{or} \quad M'(g_i) = M(g_i) - 1 \quad \text{and} \quad M'(\widetilde{g_i}) = M(\widetilde{g_i}) + 1.$$

Transition t being enabled by M, for all $g_j \in Reg(i)$, the marking of place g_j verifies $M(g_j) \in [\omega_{t,j}, Max_j - \omega'_{t,j}]$, where $\omega_{t,j}$ is the weight of the test arc connecting t to g_j (there is no arc if $\omega_{t,j} = 0$), and $\omega'_{t,j}$ is the weight of the test arc connecting t to $\widetilde{g_j}$ (there is no arc if $\omega'_{t,j} = 0$). Recall that $M(g_j)$ defines a level x_j of the component $g_j \in Reg(i)$, hence M defines a state assignment, in particular it gives the levels of the regulators of g_i. Therefore, from Definition 5, M allows us to recover a path Φ in the MDD of $\mathcal{K}_i$. The assignment of any x_j along Φ verifies: $\phi_j = \omega_{t,g_j}$. and $\phi'_j = \omega'_{t,g_j}$. Moreover, we have (because t is enabled in M):

$$t = t^+_{i,\Phi} \text{ and } \quad M(g_i) = x_i \leq v_\Phi - 1 \quad \text{(arc from } \widetilde{g_i} \text{ to } t \text{ weighted } Max_i - v_\Phi + 1),$$
$$\text{or} \quad t = t^-_{i,\Phi} \text{ and } \quad M(g_i) = x_i \geq v_\Phi + 1 \quad \text{(arc from } g_i \text{ to } t \text{ weighted } v_\Phi + 1).$$

Therefore, $v_\Phi \neq x_i$ and there exists a transition in $(\mathcal{S}, \mathcal{T})$ from state x (defined by M) to state x' such that $\forall j \in \mathcal{G}, x'_j = x_j$, and $x'_i = x_i + 1$ if $t = t^+_{i,\Phi}, x'_i = x_i - 1$ if $t = t^-{}_{i,\Phi}$. □

The MDD representation of $\mathcal{K}$ leads to more compact Petri nets compared to those obtained from decision trees (as in Chaouiya et al. 2006; see Fig. 4 for an illustration).

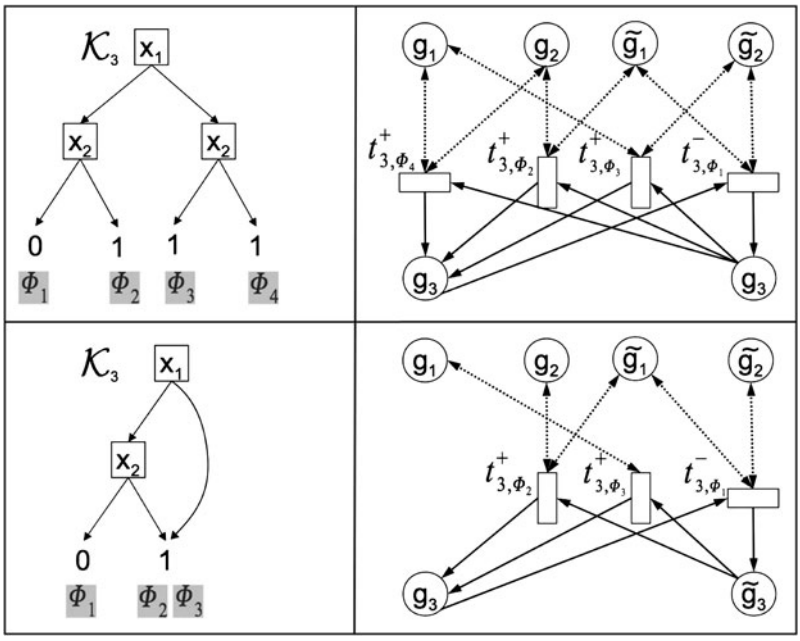


Fig. 4 Translations of decision tree versus decision diagram representations of a logical function into standard Petri nets. Here, we consider a node g_3 with two regulators g_1 and g_2. *Top*: the decision tree representing $\mathcal{K}_3$ is given with the corresponding MRPN, which encompasses one transition for each path of the tree (hence four transitions). *Bottom*: the decision diagram representing $\mathcal{K}_3$, leading to three relevant paths, and thus to three transitions in the derived MRPN

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 isomorph. This leads to the following property.

Property 2 *Given a LRG $\mathcal{R}$, two different orderings of the regulatory nodes can lead to different MRPNs, which have the same dynamical behaviour* (*i.e. their marking graphs are isomorph for a given initial state x*).

The proof easily follows from Property 1. Figure 5 displays the two MRPNs obtained from the same LRG, considering different orderings of the variables.

Two invariant properties easily follow from Definition 5. The MRPN corresponding to a LRG is covered by n P-invariants (n being the number of regulatory components). Moreover, T-invariants are always defined as pairs of transitions related to pairs of complementary places. These properties can be used as a consistency check of the MRPN. This leads to the question of a possible reverse transformation, which is not addressed here. Note that several LRGs might be recovered from a given MRPN (i.e. a PN satisfying structural constraints such as the P and T-invariants properties). This has been already emphasized for the Boolean case in Chaouiya et al. (2004). Indeed, if we consider a pair of complementary places, we have no way to determine which one corresponds to the current level of the regulatory product (see Fig. 6).

This systematic rewriting has been implemented in our software GINsim (Naldi et al. 2009a). In Chaouiya et al. (2008), the rewriting rules are demonstrated for a multi-valued

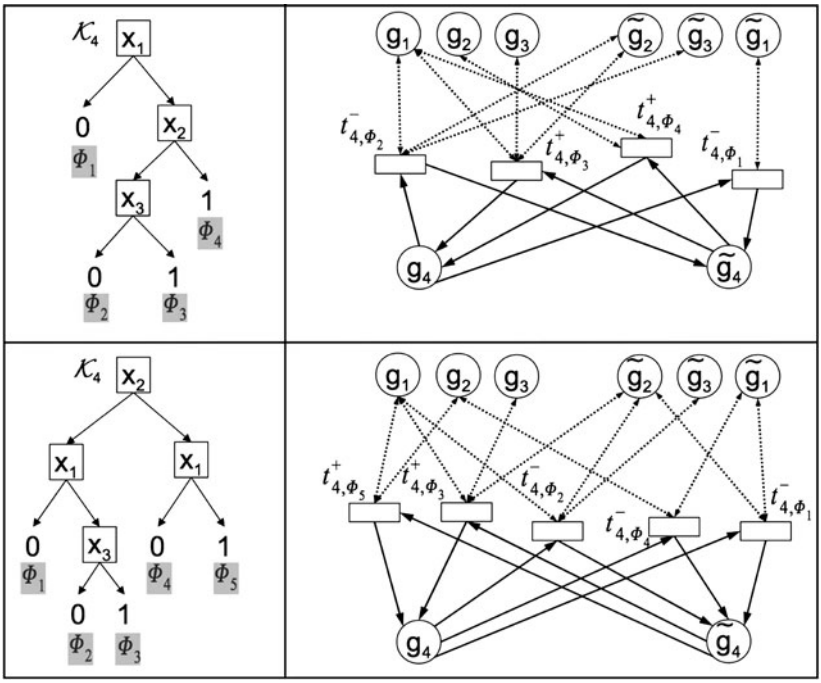


Fig. 5 Effect of decision variable ordering on the structure of the resulting MRPN. Here, we consider a node g_4 with three regulators g_1, g_2 and g_3. On the left, two decision trees are displayed, both representing $\mathcal{K}_4$. On the right, the resulting MRPNs are given. The number of paths in the decision tree determines the number of transitions in the corresponding MRPN. Here, there is a unique transition for each path since the values labelling the leaves are 0 or $Max_4 = 1$

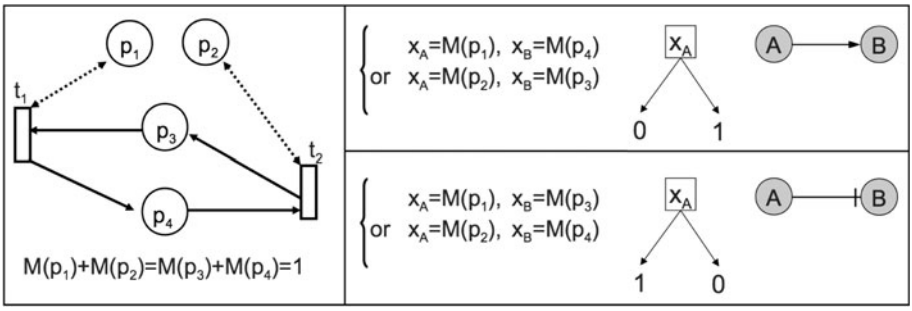


Fig. 6 Two different LRGs induced from a single MRPN-like PN. On the left, part of a MRPN is given, with two pairs of complementary places (p_1, p_2) and (p_3, p_4), defining two regulatory nodes A and B. On the right, the four possible choices for the places representing the current levels of A and B, and the corresponding decision trees and associated regulatory structures (activation vs. inhibition). For example (top case), if x_A is given by the marking of p_1 and x_B by the marking of p_4, then the network structure indicates that when $x_A = 1$, if $x_B = 0$ (i.e. $M(p_3) = 1$) transition t_1 may fire and increase the level of B. In other words, this choice denotes a situation where A is an activator of B

logical model of the genetic switch controlling the lysis-lysogeny decision in the bacteriophage lambda. This and other PN models can be downloaded from the GINsim web site, as text files in the INA format.

3.2 Coloured Petri net representation

MRPNs might seem complex and the regulatory structures underlying the model are not easy to visualize. In this section, we present a coloured version of MRPNs, called Coloured Regulatory Petri Nets (CRPNs), where there is one place for each regulatory component g_i, and one transition governing the evolution of the marking $M(g_i)$ (for further details, see Chaouiya et al. 2006).

Definition 6 A LRG $\mathcal{R} = (\mathcal{G}, \mathcal{M}ax, \mathcal{E}, \Theta, \mathcal{K})$ can be represented as a Coloured Regulatory Petri Net (CRPN), with, for all $g_i \in \mathcal{G}$:

- one place g_i containing a unique token, which value is the current level x_i of g_i ($x_i \in [0, Max_i]$);
- one transition t_i connected:
  - to each place g_j such that $g_j \in Reg(i)$, by tests arcs labelled with the arc expression x_j;
  - to place g_i, by an incoming arc labelled with the arc expression x_i, and by an outgoing arc labelled with the arc expression:[3] $x_i + sign(\mathcal{K}_i(x) - x_i)$;
- a guard associated to transition t_i, to ensure that t_i is enabled only if g_i is called to change its current value, hence this guard is defined as $x_i \neq \mathcal{K}_i(x)$.

Figure 7 illustrates the CRPN representing the logical regulation of a gene.

A property similar to Property 1 can be stated, equating the LRG state transition graph to the related CRPN marking graph.

Finally, contrary to MRPNs, a unique LRG can be recovered from a given CRPN.

4 Introducing priorities

For specific initial conditions, the asynchronous dynamics of realistic regulatory graphs often leads to huge numbers of states. In the logical framework, as introduced in Sect. 2, we consider asynchronous updatings for the definition of state transition graphs, which match the corresponding MRPN marking graphs. Another updating policy frequently considered consists in applying all updating calls at once (Kauffman 1993). In Steggles et al. (2007), the authors propose a PN rewriting of Boolean regulatory networks with such a synchronous updating. Although generating simpler, deterministic state transition graphs, the synchronous updating often generates spurious behaviours (Thomas 1991). In contrast, since the asynchronous updating makes no assumption on the delays related to the increase or decrease orders, the generated dynamics is highly non-deterministic and realistic trajectories are hidden by numerous unrealistic ones. The inclusion of qualitative delays in logical models has been considered by several authors (e.g. Ahmad et al. 2006; Siebert and Bockmayr 2006, 2007; Thomas 1991). Siebert and Bockmayr (2007) pointed out that delays for synthesis or decay of a regulatory product might be context sensitive. Sound delays are generally difficult to obtain from experimental data. Moreover, other questions arise in case of preemption: when a process (synthesis or decay) is interrupted for a while, which delay value should be considered when the process resumes?

[3] For $x \in \mathbb{Z}, sign(x) = +1$ if $x > 0$, -1 if $x < 0$, 0 otherwise.

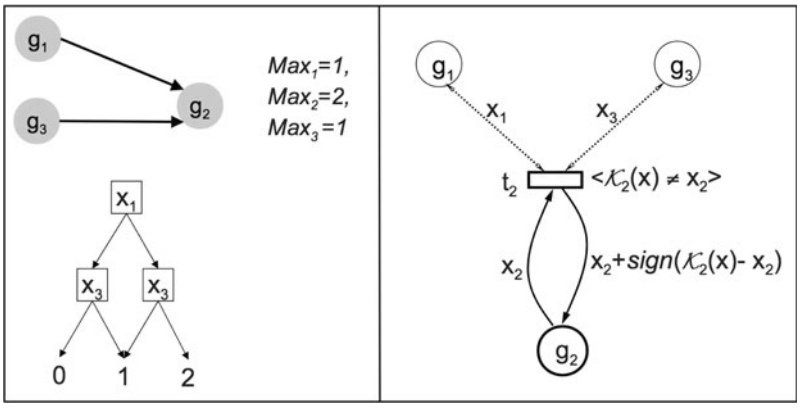


Fig. 7 Coloured Petri net representation of a LRG. In the simple case depicted here, the unique transition t_2 that governs the behaviour of g_2 is connected to the regulators of g_2 by test arcs, which read the current values of g_1 and g_3. A guard associated to t_2 ensures that the current level of g_2 (x_2) is different from its target value $\mathcal{K}_2(x)$. When t_2 is enabled, it increases or decreases the value x_2 by one (according to the comparison of x_2 and $\mathcal{K}_2(x)$). See Fig. 2, bottom panel, for a "flat" (MRPN) version of this model

To circumvent these difficult questions, we chose a simple strategy consisting in sorting trajectories through the introduction of priority classes (Fauré et al. 2006). This possibility has been introduced in GINsim by allowing the user to group components into different classes, and to assign a priority level to each of these classes. In case of concurrent transition calls, GINsim only updates the component(s) belonging to the class with the highest ranking. For each regulatory component class, the user can further specify the desired updating assumption (synchronous or asynchronous), which then determines the treatment of concurrent transition calls inside that class. When several classes have the same ranking, concurrent transitions are treated under an asynchronous assumption (no priority). Moreover, similarly to the definitions of transitions t^+ and t^- in Definition 5, one can distinguish between increasing and decreasing tendencies when updating node levels. In biological terms, increasing (respectively decreasing) tendencies generally correspond to synthesis (respectively degradation) processes. GINsim further enables the distinction between these two types of updatings.

Introduction of priorities in MRPNs is straighforward, if we only consider asynchronous classes. Taking into account asynchronous classes in a MRPN simply consists in defining a *priority function* that assigns a positive integer to each transition defining its priority level (Marsan et al. 1994). As priorities restrict the enabling of transitions, it is clear that the marking graph of the prioritised model is a subgraph of the marking graph of the original model. Hence, we must be aware that the introduction of priorities in a model may affect the reachability of attractors, and may even result to additional or modified attractors. Figure 8 illustrates how prioritisation may affect the configuration of attractors.

The use of priority classes eases the analysis of large regulatory networks (see, e.g., Sánchez et al. 2008). Priorities can be set on the basis of biological knowledge. Hence, the paths lost in the marking graphs are arguably not realistic. It could be interesting to define subtler classes, for example depending not only on the sign of the update, but also on the combination of interactions leading to this update. In this case, it would be necessary to impede the automatic simplification of the MRPN (as in Fig. 4), because this amounts in grouping several situations, which could then belong to distinct priority classes.

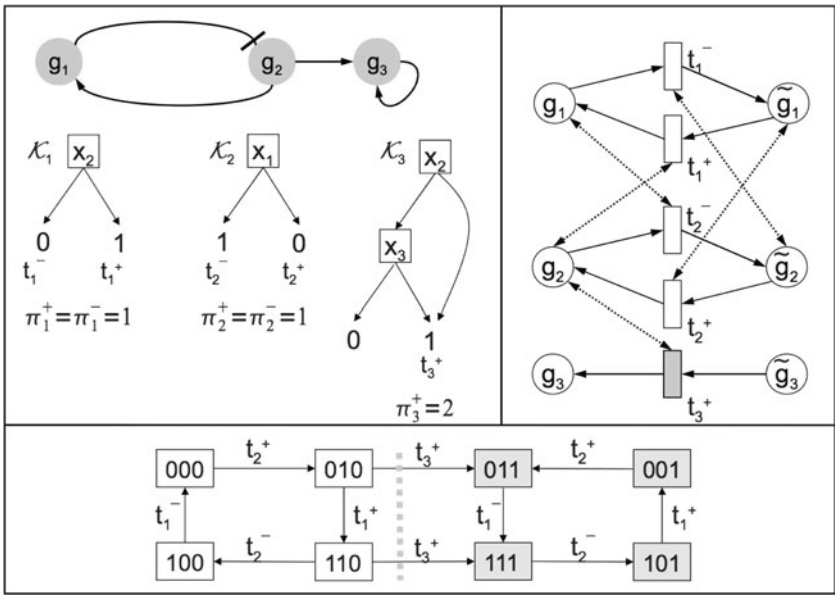


Fig. 8 Effect of prioritisation on the dynamics. The *top left panel* shows a LRG with the specification of two priority classes: all updating calls for g_1 and g_2 have a higher priority ($\pi_1 = \pi_2 = 1$), g_3 has a lower priority ($\pi_3 = 2$). The *top right panel* displays the corresponding MRPN, and transition t_3^+ that has a lower priority is greyed out. The marking graph is given in the *bottom panel*, generated from the initial state (0, 0, 0). If priorities are implemented, the dynamics is restricted to the left-hand cycle, otherwise it will end up in the rightmost terminal cycle. The transition from the first cycle to the terminal cycle is only possible through t_3^+, which is always in conflict with another transition with a higher priority. However, the right hand cycle is still an attractor for the prioritised model, if one considers an initial marking in this cycle

5 Feedback circuits and functionality context analysis

For complex regulatory networks, R. Thomas enunciated several rules binding the dynamical behaviour to the presence of specific types of regulatory circuits. More precisely, he has conjectured that a necessary condition for multistationarity is the presence of a positive circuit (i.e., containing an even number of inhibitions), whereas a necessary condition for homeostasis and/or sustained, stable oscillations is the presence of a negative circuit (with an odd number of inhibitions), cf. Thieffry (2007), Thomas and D'Ari (1990), Thomas et al. (1995) and references therein. These rules inspired several theorems refering to different frameworks (see Remy et al. 2006a; Richard and Comet 2007; Soulé 2006 and references therein).

However, the presence of a regulatory circuit is not sufficient to enable the corresponding dynamical property. The simple example presented in Fig. 9 illustrates this point. When g_3 is present ($x_3 = 1$), the negative circuit involving g_1 and g_2 induces a cycle in the state transition graph; when g_3 is absent ($x_3 = 0$), the cyclic behaviour disappears: the circuit is no longer *functional*. A circuit is said to be *functional* if it does generate homeostasis in the case of a negative circuit, or multistationarity in the case of a positive circuit. In Naldi et al. (2007), we associated a functionality context to each circuit of a regulatory graph. When a circuit is embedded in a regulatory graph, its (external) inputs may annihilate the expected property (as illustrates in Fig. 9). The functionality context of

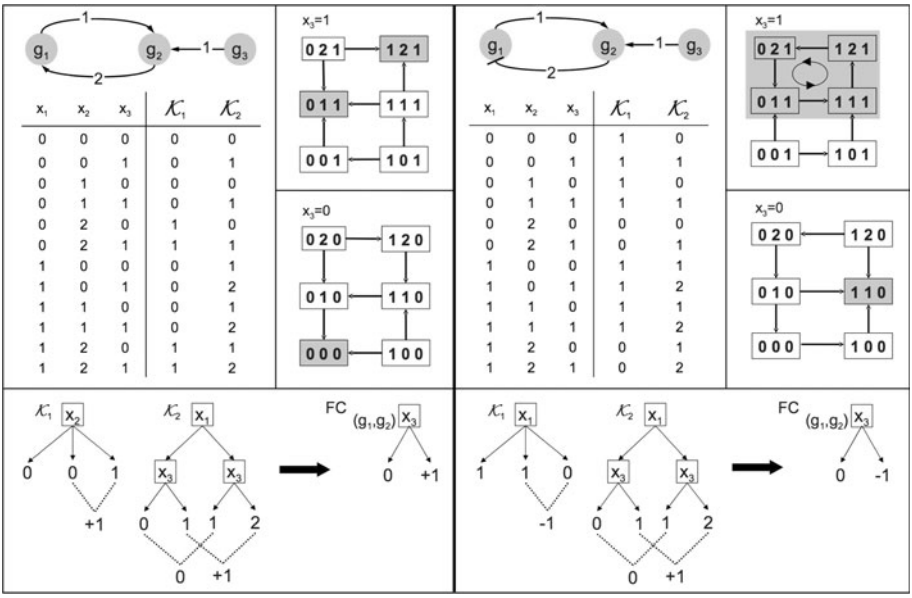


Fig. 9 Functionality context of regulatory circuits. An example of positive circuit is displayed on the *left*, along with a negative circuit on the *right*. For each case, the *upper-left panel* displays the circuit and the table defining $\mathcal{K}_1$ and $\mathcal{K}_2$. The state transition graph in presence (*resp.* absence) of g_3 is shown in the *upper-right* (*resp. middle-right*) panel. The *bottom panels* show the decision trees representing the functions $\mathcal{K}_i$ and their use to determine the functionality context of the circuit. For example, the functionality context of the interaction (1, 2, 1) within the circuit is determined from the decision tree of $\mathcal{K}_2$ by comparing the values of leaves reached from $x_1 = 0$ (absence of g_1) with those reached from $x_1 = 1$ (illustrated by the *dotted lines* below the trees). In both circuits, the interaction (2, 1, 2) is always functional. The interaction (1, 2, 1) is functional within the circuit only in the presence of g_3. Indeed, $\mathcal{K}_2(0,0) = 0$ and $\mathcal{K}_2(1,0) = 1$: even if g_1 has a positive effect on g_2, in the absence of g_3, this effect is not sufficient to make g_2 cross the threshold of the next interaction. The context of the whole circuit is then obtained by combining those of the individual interactions. The state transition graphs are consistent with these results: for the positive (*resp.* negative) circuit, multistability (*resp.* oscillations) appears only for $x_3 = 1$. Attractors (stable states and attracting cycles) are emphasized in *grey*

a circuit defines constraints on the levels of external regulators of the circuit enabling the circuit to be functional.

In what follows, we propose a formal definition for regulatory circuits and their functionality contexts along with a method to determine these contexts.

Definition 7 A circuit $\mathcal{C} = \{(i_l, i_{l+1}, \theta_{l,l+1}), l = 1, \ldots, k\}$ of a regulatory graph $\mathcal{R} = (\mathcal{G}, \mathcal{M}ax, \mathcal{E}, \Theta, \mathcal{K})$ is a subgraph such that $\{i_1, \ldots, i_k\} \in \mathcal{G}$ and $(i_l, i_{l+1}, \theta_{l,l+1}) \in \mathcal{E}$ for $l = 1,\ldots k$ with the convention $k + 1 = 1$.

The following definition formalises the notion of functionality of an interaction within a regulatory circuit.

Definition 8 An interaction (i_l, i_{l+1}, $\theta_{l,l+1}$) of a circuit $\mathcal{C}$ is functional if and only if there exists $x \in \mathcal{S}$ with $x_{i_l} = \theta_{l,l+1} - 1$ and $x' \in \mathcal{S}$ with $x'_{i_l} = \theta_{l,l+1}$ and $x'_k = x_k$ for all $k \neq i_l$, such that:

$$\mathcal{K}_{i_{l+1}}(x) < \theta_{l+1,l+2} \leq \mathcal{K}_{i_{l+1}}(x'),$$
$$\text{or } \mathcal{K}_{i_{l+1}}(x') < \theta_{l+1,l+2} \leq \mathcal{K}_{i_{l+1}}(x).$$

Definition 8 establishes that the interaction $(i_l, i_{l+1}, \theta_{l,l+1})$ is functional provided its activity affects the activity of the following interaction of the circuit (going out i_{l+1}). This depends on the values of $\mathcal{K}_{l+1}$, considering values $\theta_{i_l,i_{l+1}} - 1$ and $\theta_{i_l,i_{l+1}}$ for i_l and all possible values of other regulators of i_{l+1}.

We can then define the sign of an interaction: when the increase of the source across its threshold drives an increase (*resp.* decrease) of the target across the threshold of the following interaction, the interaction is functional and positive (value + 1) (*resp.* negative, value − 1), otherwise the interaction is not functional (value 0).

From Definition 8, one can determine the set of assignments for the regulators of i_{l+1} (except for i_l) for which the interaction $(i_l, i_{l+1}, \theta_{l,l+1})$ is functional (its sign is not 0). This gives the context of functionality of the interaction.

Definition 9 The functionality context of a circuit $\mathcal{C}$ is given by the intersection of the functionality contexts of its interactions.

Note that if the functionality context is empty, then the circuit is not functional.

In Naldi et al. (2007), we define a logical function that yields the sign of the interaction targeting i_{l+1}. This function is again represented as a MDD and is obtained from the logical function $\mathcal{K}_{l+1}$. The functionality context of a circuit is defined as the intersection of the contexts of its constitutive interactions. Hence, combining the functionality conditions of all the interactions of a circuit in a logical conjunction, we obtain the functionality context of the whole circuit.

Summarising, the computational method given in Naldi et al. (2007) allows the determination of the functionality contexts (and signs) of a circuit (notice that, depending on the values of its regulators, the sign of a regulatory circuit might change). The procedure involves two steps:

- the determination of the sign of each interaction, using decision diagrams,
- the computation of the product of these values by combining these MDDs.

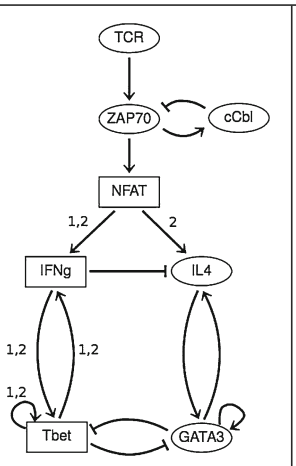
In the resulting decision diagram, the paths leading to non-zero leaves define the functionality context of the circuit.

In Remy et al. (2006b), we defined how to determine, in Boolean Regulatory PNs (BRPNs), the functionality contexts of regulatory circuits. The method relies on the comparison of the effects of relevant pairs of transitions along the circuit. This comparision is performed by analysing the matrices *Pre* and *Post* of the net (see Sect. 6 for an illustration). The proposed procedure is based on the analysis of large matrices and, more important, is only valid for Boolean models. Its extension to the multi-valued case is not straighforward. However, this seminal work inspired the principles driving the computational method mentioned above, which is based on MDDs manipulations. Hence, currently we analyse the functionality contexts of regulatory circuits in the logical framework, using the MDD representation of the logical functions. The algorithm has been implemented into GINsim and efficiently determines the circuit functionality contexts for large LRGs (including multi-valued cases). In principle, this algorithm could be adapted to the analysis of regulatory circuits in CRPNs as a unique transition is associated to each regulatory component carrying its logical function.

6 Biological illustration: a simplified qualitative model for T cell activation and differentiation

6.1 Logical model

T lymphocytes play a central role in the regulation of the adaptive immune response. Early differentiation steps in the thymus lead to a pool of antigen-naive T helper cells (denoted Th0), displaying different antigen-specific T Cell Receptor (TCR). Once in the periphery, when a Th0 cell encounters a matching antigen (displayed by an antigen presenting cell), the activation of the TCR receptor triggers a cascade of events ultimately leading to the activation of the cell and cell differentiation. Th0 cells can then differentiate into Th1 or Th2 subtypes, which enhance different (cellular vs. humoral) immune responses. To illustrate the definitions and methods introduced above, we introduce a model (see Fig. 10) integrating two recently published models. The first of these models encompasses 45 components transducing the signals received by the TCR and two co-receptors on the cell membrane down to transcription factors in the nucleus (Klamt et al. 2006). Involving 17 components, the second model focuses on the control of the differentiation of Th0 cells into Th1 vs. Th2 subtypes, characterised by the activation of Tbet and the secretion of IFNγ, versus the activation of GATA3 and the secretion of IL4, respectively (Mendoza 2006). These two models share only two components (TCR and NFAT) and can thus be easily coupled using the logical formalism. The coupled model was then reduced (in part automatically with a novel GINsim prototype, in part manually). We retained only eight



$Node$	$Value$	$Function$
TCR	–	–
$ZAP70$	1	$TCR \wedge \neg cCbl$
$cCbl$	1	$ZAP70$
$NFAT$	2	$ZAP70$
IFNg	1	$(NFAT_1 \wedge \neg Tbet) \vee (Tbet_1 \wedge \neg NFAT)$
	2	$(NFAT_1 \wedge Tbet_1) \vee NFAT_2 \vee Tbet_2$
Tbet	1	$\neg GATA3 \wedge (Tbet_1 \vee IFNg_1) \wedge \neg(Tbet_2 \vee IFNg_2)$
	2	$\neg GATA3 \wedge (Tbet_2 \vee IFNg_2)$
$IL4$	1	$NFAT_2 \vee GATA3$
$GATA3$	1	$(GATA3 \vee IL4) \wedge \neg Tbet$

Fig. 10 Reduced logical model for T cell activation and differentiation. On the *left*, the regulatory graph is displayed. *Blunt arrows* denote inhibitions while regular ones denote activations. *Circled components* are represented by Boolean variables, while *rectangular boxes* emphasise ternary components. Interaction thresholds are specified on the drawing if different from 1 (1, 2 denote two thresholds for multi-arcs). The table on the *right* gives the logical rules for each component of the model. For the sake of readability, the logical functions are displayed as logical statements, using the classical connectors ($\neg, \wedge, \vee$ denoting NOT, AND and OR operators, respectively). *TCR* stands for $TCR = 1$, $Tbet_1$ and $Tbet_2$ stand for $Tbet = 1$ and $Tbet = 2$, respectively (the same for remaining variables). The rules are given for the non-zero target values (target value is 0 for the complementary statement)

interacting components (out of 60), selected to preserve the main dynamical properties of the original models:

- the transient oscillations of ZAP70 and cCbl upon T cell activation;
- the coexistence of four stable states accounting for different T cell populations.

For proper parameterisation, this logical model has four stable states, corresponding to the following cell types:

- Th0: all components are inactive;
- Th1: Tbet and IFNγ are active at medium level (1);
- Th1*: Tbet and IFNγ are active at their highest level (2);
- Th2: GATA3 and IL4 are active.

Starting from the Th0 state, a transient activation of the TCR leads to a transient activation of NFAT. Under constant TCR activation, the level of NFAT oscillates with that of ZAP70, which is involved in a negative circuit with cCbl. The activations of IFNγ and IL4 are coupled to that of NFAT. These two cytokines in turn control the activation of Tbet and GATA3. Self-regulations and cross-inhibitions of these transcription factors constitute the switch enabling a stable differentiation of Th1 and Th2 subtypes.

The model encompasses 11 regulatory circuits. Five of these circuits play crucial roles:

- The negative circuit involving ZAP70 and cCbl is functional when the TCR is active. In this context, it triggers oscillations of the activity levels of these two components, further driving oscillations of downstream targets.
- The positive, cross-inhibitory circuit involving GATA3 and Tbet is functional in the presence of IFNγ and IL4. This circuit prevents the cell from reaching a chimeric state.
- The positive circuit involving GATA3 is functional in the absence of Tbet and IL4.
- The positive circuit involving medium Tbet is functional in the absence of GATA3 and IFNγ.
- The positive circuit involving high Tbet is functional in the absence of GATA3 and for low or medium IFNγ.

The auto-regulations of GATA3 and Tbet enable the memorisation of their activation (maintenance) above the corresponding thresholds. However, the coexistence of Tbet and GATA3 is forbidden by their cross-inhibitory effects, thus leading to four stable states characterised by the exclusive expression of one of these factors, or yet of none of them.

6.2 MRPN representation

We have generated the MRPN corresponding to the LRG as defined in Fig. 10. It encompasses 16 places and is covered by 8 P-invariants, which correspond to the 8 pairs of complementary places (one pair for each regulatory component):

$$\begin{aligned} M(IFNg) + M(\widetilde{IFNg}) &= 2, \quad M(Tbet) + M(\widetilde{Tbet}) = 2, \\ M(IL4) + M(\widetilde{IL4}) &= 1, \quad M(GATA3) + M(\widetilde{GATA3}) = 1, \\ M(cCbl) + M(\widetilde{cCbl}) &= 1, \quad M(ZAP70) + M(\widetilde{ZAP70}) = 1, \\ M(NFAT) + M(\widetilde{NFAT}) &= 2, \quad M(TCR) + M(\widetilde{TCR}) = 1. \end{aligned}$$

As previously mentioned, the number of transitions can vary, depending on the ordering of the variables (see Fig. 5). For example, considering the ordering (*IFN*γ, *Tbet*, *IL4*,

*GATA*3, *cCbl*, *ZAP*70, *NFAT*, *TCR*), the MRPN has 28 transitions and 16 T-invariants. In this case, 7 transitions govern the evolution of *Tbet*. Now, if we change the ordering to (*GATA*3, *IFN*γ, *Tbet*, *IL4*, *cCbl*, *ZAP*70, *NFAT*, *TCR*), the MRPN has 25 transitions and 7 T-invariants, and only 3 transitions governing *Tbet* evolution. This raises the question of finding an optimal ordering for each LRG node, seeking a lower number of transitions.

Figure 11 illustrates the part of the MRPN dealing with the regulation of *Tbet*. Here, we have two T-invariants: $(t^+_{\Phi_1}, t^-_{\Phi_3})$ and $(t^+_{\Phi_2}, t^-_{\Phi_3})$.

We have checked that, with an initial marking corresponding to a level 1 for *TCR* and 0 for all other components, the marking graph encompasses 336 markings, including four dead markings corresponding to the four cellular states described above.

Focusing on the Boolean regulatory circuit between *cCbl* and *ZAP*70, we now exemplify the determination of functionality contexts (see Fig. 12). First, we apply the method proposed in Remy et al. (2006b), which applies only in the Boolean case and consists in comparing the effects of pairs of transitions related to the components involved in the circuit (see Remy et al. 2006b for further details). The relevant pairs of transitions are determined from both the matrix *Pre* (giving the arcs between places and transitions) and the matrix *Post* (giving the arcs from transitions to places).

For example, to determine the functionality context of the interaction from *cCbl* towards *ZAP*70, we have to check if it has an effect on *ZAP*70 for fixed levels of *TCR* (the only external regulator of the circuit, acting on *ZAP*70). For this purpose, we select the pairs of transitions (t, t') such that:

1. $Pre(ZAP70, t) - Post(ZAP70, t) \neq 0$ and $Pre(ZAP70, t') - Post(ZAP70, t') \neq 0$: both t and t' change the value of *ZAP*70;
2. $Pre(cCbl, t) \neq Pre(cCbl, t')$: t and t' account for different constraints on *cCbl* (presence vs. absence of *cCbl*);
3. $Pre(TCR, t) = Pre(TCR, t')$ or $Pre(\widetilde{TCR}, t) = Pre(\widetilde{TCR}, t')$: t and t' account for compatible constraints on *TCR*.

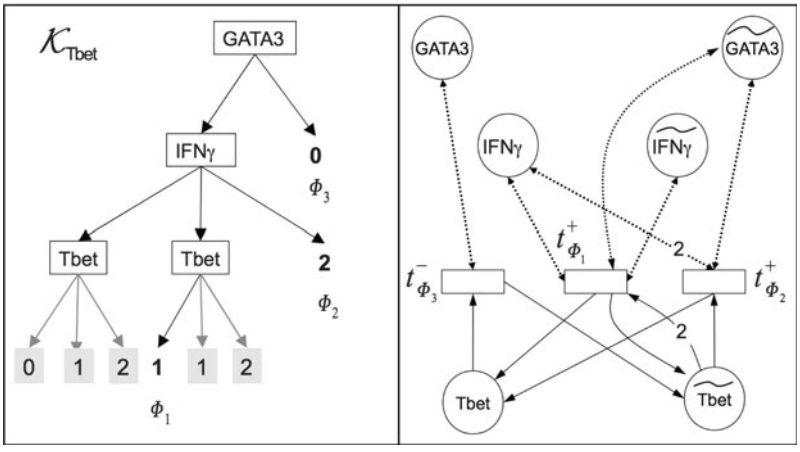
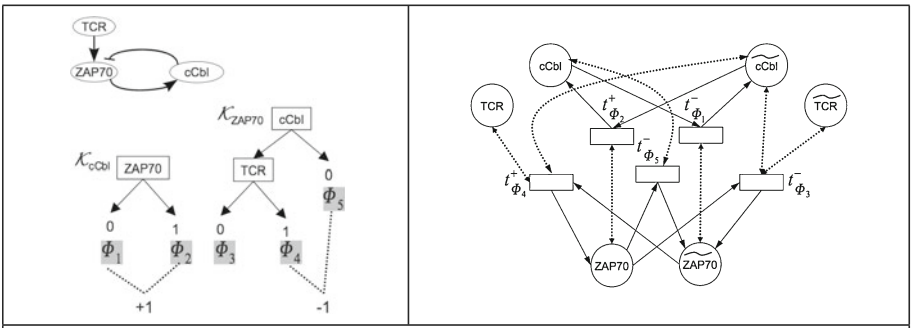


Fig. 11 Determination of the MRPN representation for *Tbet*. The *left panel* gives the MDD for $\mathcal{K}_{Tbet}$, which governs the behaviour of *Tbet*. Path names label relevant leaves (Φ_1, Φ_2 and Φ_3), whereas leaves corresponding to situations where *Tbet* is not called to change (because of its self-regulation) are greyed out. The *right panel* illustrates the MRPN obtained for the regulation rules of *Tbet*, with three transitions



$Pre =$

	$t^-_{\Phi_1}$	$t^+_{\Phi_2}$	$t^-_{\Phi_3}$	$t^+_{\Phi_4}$	$t^-_{\Phi_5}$
$cCbl$	**1**	**0**	0	**0**	**1**
$\widetilde{cCbl}$	0	1	1	1	0
$ZAP70$	**0**	**1**	1	**0**	**1**
$\widetilde{ZAP70}$	1	0	0	1	0
TCR	**0**	**0**	0	**1**	**0**
$\widetilde{TCR}$	**0**	**0**	1	**0**	**0**

$Post =$

	$t^-_{\Phi_1}$	$t^+_{\Phi_2}$	$t^-_{\Phi_3}$	$t^+_{\Phi_4}$	$t^-_{\Phi_5}$
$cCbl$	**0**	**1**	0	0	1
$\widetilde{cCbl}$	1	0	1	1	0
$ZAP70$	0	1	0	**1**	**0**
$\widetilde{ZAP70}$	1	0	1	0	1
TCR	0	0	0	1	0
$\widetilde{TCR}$	0	0	1	0	0

Fig. 12 Analysis of the *ZAP*70-*cCbl* circuit. The *top left panel* gives the Boolean circuit encompassing *ZAP*70 and *cCbl* in the LRG of Fig. 10. The MDDs associated to functions $\mathcal{K}$ are shown. On these MDDs, we can verify that the interaction from *ZAP*70 to *cCbl* is always functional and positive, whereas the interaction from *cCbl* to *ZAP*70 is functional and negative in the presence of *TCR* (*dotted lines* join the situations that have to be compared). The *top right panel* displays the corresponding MRPN. The *bottom panel* shows the submatrices *Pre* and *Post* for the three relevant nodes: *cCbl*, *ZAP*70 and *TCR*. Using these matrices, we can check the functionality context of the interaction from *cCbl* to *ZAP*70: we first select $t^-_{\Phi_3}, t^+_{\Phi_4}$ and $t^-_{\Phi_5}$, which change the marking of *ZAP*70. Then, we consider the pair $(t^+_{\Phi_4}, t^-_{\Phi_5})$ because these transitions are differently constrained by *cClb* and have different effects on *ZAP*70 (in contrast with the pair $(t^-_{\Phi_3}, t^-_{\Phi_5})$). Since $t^+_{\Phi_4}$ requires the presence of *TCR*, whereas $t^+_{\Phi_5}$ is not constrained by *TCR*, we conclude that the interaction is functional in the presence of *TCR*. The interaction from *ZAP*70 to *cCbl* is always functional. Elements in bold in the matrices allow us to select the relevant pairs of transitions and to conclude on the functionality of the interactions

For such a pair (t, t'), we further check if $Post(t, ZAP70) \neq Post(t', ZAP70)$. If it is the case, the interaction is functional (the effect on *ZAP*70 is different) for specific values of *TCR* (given by the constraint on *TCR* for both t and t').

Here, the pair of transitions satisfying the conditions listed above is $(t^+_{\Phi_4}, t^-_{\Phi_5})$ (see bold elements in the matrices in Fig. 12). Since $Pre(TCR, t^+_{\Phi_4}) = 1$, the interaction from *cCbl* towards *ZAP*70 is functional in the presence of *TCR*.

Figure 12 displays the relevant part of the MRPN encompassing the regulatory circuit between *cCbl* and *ZAP*70, subject to a regulation by *TCR*. The MDDs are given for $\mathcal{K}_{cCbl}$ and $\mathcal{K}_{ZAP70}$ and, as in Fig. 9, dotted lines connect the values that must be compared, and the sign of the interaction is given. This illustrates how the functionality contexts are computed from the MDDs representing the logical functions. Figure 12 also shows the *Pre* and *Post* submatrices relevant for the analysis of the *ZAP*70 − *cCbl* circuit.

7 Discussion

This article reviews a series of results allowing the translation of logical regulatory models into discrete Petri nets and, thereby, the combination of complementary analytical methods

and computational tools (reachability analysis, PN invariants, regulatory circuit analysis, etc.).

A different rewriting of LRGs into Coloured PNs (CPNs) is proposed in Comet et al. (2000), which comprises just one place (accounting for the whole state of the LRG) and one transition (accounting for the dynamics of the LRG). Leaning on this CPN representation, the authors further propose a method to automatically generate sets of logical rules (or *logical parameters*) compatible with the topology of a regulatory graph and temporal logic formulae capturing dynamical properties of the corresponding biological system.

Here, we lean on CPN rewriting to further transpose one of the most original aspect of the logical modelling method, namely the analysis of the dynamical roles of the regulatory circuits embedded in complex networks. For a given CRPN, this approach enables the identification of positive or negative circuits at the basis of multistationary properties or sustained oscillatory behaviour, respectively. This is facilitated by the fact that in CRPNs, for each LRG node there is one transition governing its behaviour, contrary to the rewriting proposed in Comet et al. (2000), where a unique transition accounts for the behaviours of all the nodes. The proposed method enables the delineation of definite constraints on the marking of the places. These constraints are derived from the logical rules associated with the transitions feeding the places involved in a circuit. However, further work is required to clarify how interconnected circuits can cooperate to generate more complex behaviours.

We have previously shown how the combination of logical and PN formalisms can be applied to the dynamical modelling and analysis of regulated metabolic pathways (Simão et al. 2005; see Sackmann et al. 2006 for a complementary approach). In this context, the circuit analysis delineated here complements existing PN methods (e.g. computation of dead markings, P- and T-invariants, reachability analysis) to better cope with regulated metabolic networks, i.e. by providing an abstraction, which is relevant from a dynamical point of view.

We have illustrated PN rewriting using a simplified logical model for the activation and differentiation of T lymphocytes. We are currently developing a full-fledged model encompassing detailed signalling transduction cascades and additional differentiation pathways. The analysis of the resulting network, which currently involves over 60 regulatory components, will clearly benefits from the exploitation of the full set of tools associated with logical (LRG) and PN frameworks. In particular, adequate model reduction methods should help to determine important dynamical properties for large networks (cf. Naldi et al. (2009b) for a first step in this direction, using the logical framework).

Although time is often implicitely considered in logical models, the use of priorities or time delays enables the specification of temporal constraints associated with concurrent processes (cf. Sect. 4). In this respect, the PN framework offers a variety of refined temporisation methods, from the definition of delay intervals to stochastic firing laws (cf. Li et al. (2007) for an application of timed-PN to model a biological signalling pathway, and Mura and Csikasz-Nagy (2008) for a SPN modelling the yeast cell cycle control).

The modelling and analysis of complex biological regulatory networks implies the capacity to handle and combine different levels of abstractions or details, to identify and compose relevant network modules, and to delineate the associated essential dynamical properties. Whenever experimental data are sufficiently abundant and precise, qualitative PN models (or sub-models) can be refined to generate predictive quantitative models, taking advantage of timed, stochastic, continuous, or hybrid PN versions (cf., e.g., Srivastava et al. 2001; Nagasaki et al. 2004; Doi et al. 2006; Li et al. 2007; Mura and Csikasz-Nagy 2008; Heiner et al. 2008). In this respect, the progressive integration of

methods transposed from other dynamical modelling frameworks and newly developed PN analysis tools should ease the definition of hierarchical or modular models (see e.g. Grafahrend-Belau et al. 2008; Sackmann et al. 2006).

Acknowledgments A.N. has been supported by a PhD grant from the French Ministry of Research and Technology. C.C. acknowledges the support provided by the Calouste Gulbenkian Foundation. This work was further supported by research grants from the French National Agency (projects ANR-06-BYOS-0006 and ANR-08-SYSC-003), and from the Belgian Science Policy Office (IAP BioMaGNet).

References

Ahmad J, Richard A, Bernot G, Comet J-P, Roux O (2006) Delays in biological regulatory networks (BRN). Lect Notes Comput Sci 3992:887–894

Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P (2008) Molecular biology of the cell, 5th edn. Garland Science/Taylor & Francis, New York

Chaouiya C, Remy E, Mossé B, Thieffry D (2003) Qualitative analysis of regulatory graphs: a computational tool based on a discrete formal framework. Lect Notes Control Inf Sci 294:119–126

Chaouiya C, Remy E, Ruet P, Thieffry D (2004) Qualitative modelling of genetic networks: from logical regulatory graphs to standard Petri nets. Lect Notes Comput Sci 3099:137–156

Chaouiya C, Remy E, Thieffry D (2006) Qualitative Petri net modelling of genetic networks. Lect Notes Comput Sci 4220:95–112

Chaouiya C, Remy E, Thieffry D (2008) Petri net modelling of biological regulatory networks. J Discrete Algorithms 6(2):165–177

Comet J-P, Klaudel H, Liauzu S (2005) Modeling multi-valued genetic regulatory networks using high-level Petri nets. Lect Notes Comput Sci 3536:208–227

de Jong H (2002) Modeling and simulation of genetic regulatory systems: a literature review. J Comput Biol 1:67–103

Doi A, Nagasaki M, Matsuno H, Miyano S (2006) Simulation-based validation of the p53 transcriptional activity with hybrid functional Petri net. In Silico Biol 6:1–13

Fauré A, Naldi A, Chaouiya C, Thieffry D (2006) Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. Bioinformatics 22:124–131

Fauré A, Naldi A, Lopez F, Chaouiya C, Ciliberto A, Thieffry D (2009) Modular logical modelling of the budding yeast cell cycle. Mol Biosyst 5:1787–1796

Garg A, Xenarios I, Mendoza L, De Micheli G (2007) An efficient method for dynamic analysis of gene regulatory networks and in-silico gene perturbation experiments. Lect Notes Comput Sci 4453:62–76

GINsim web page: http://gin.univ-mrs.fr/GINsim/

González A, Chaouiya C, Thieffry D (2008) Logical modelling of the role of the Hh pathway in the patterning of the Drosophila wing disc. Bioinformatics 24:i234–i240

Goss PJ, Peccoud J (1998) Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets. Proc Natl Acad Sci USA 95:6750–6755

Grafahrend-Belau E, Schreiber F, Heiner M, Sackmann A, Junker BH, Grunwald S, Speer A, Winder K, Koch I (2008) Modularization of biochemical networks based on classification of Petri net t-invariants. BMC Bioinformatics 9:90

Heiner M, Gilbert D, Donaldson R (2008) Petri nets for systems and synthetic biology. Lect Notes Comput Sci 5016:215–264

INA, Integrated Net Analyzer, tool for the analysis of (Coloured) PNs: http://www.informatik.hu-berlin.de/~starke/ina.html

Kam T, Villa T, Brayton RK, Sangiovanni-Vincentelli AL (1998) Multi-valued decision diagrams: theory and applications. Int J Multiple-Valued Logic 4:9–62

Kauffman S (1993) The origins of order: self-organization and selection in evolution. Oxford University Press, New York

Klamt S, Saez-Rodriguez J, Lindquist JA, Simeoni L, Gilles ED (2006) A methodology for the structural and functional analysis of signaling and regulatory networks. BMC Bioinformatics 7:56

Li C, Ge QW, Nakata M, Matsuno H, Miyano S (2007) Modelling and simulation of signal transductions in an apoptosis pathway by using timed Petri nets. J Biosci 32:113–127

Marsan MA, Balbo G, Conte G, Donatelli S, Franceschinis G (1994) Modelling with generalized stochastic Petri nets. Wiley, New York

Mendoza L (2006) A network model for the control of the differentiation process in Th cells. Biosystems 84:101–114
Mura I, Csikasz-Nagy A (2008) Stochastic Petri net extension of a yeast cell cycle model. J Theor Biol 254(4):850–860
Nagasaki M, Doi A, Matsuno H, Miyano S (2004) A versatile Petri net based architecture for modeling and simulation of complex biological processes. Genome Inform 15:180–197
Naldi A, Thieffry D, Chaouiya C (2007) Decision diagrams for the representation of logical models of regulatory networks. Lect Notes Bioinform 4695:233–247
Naldi A, Berenguier D, Fauré A, Lopez F, Thieffry D, Chaouiya C (2009a) Logical modelling of regulatory networks with GINsim 2.3. Biosystems 97(2):134–139
Naldi A, Remy E, Thieffry D, Chaouiya C (2009b) A reduction method for logical regulatory graphs preserving essential dynamical properties. Lect Notes Bioinform 5688:266–280
Remy E, Ruet P, Thieffry D (2006a) Positive or negative regulatory circuit inference from multilevel dynamics. Lect Notes Control Inf Sci 341:263–270
Remy E, Ruet P, Mendoza L, Thieffry D, Chaouiya C (2006b) From logical regulatory graphs to standard Petri nets: dynamical roles and functionality of feedback circuits. Lect Notes Comput Sci 4230:55–72
Richard A, Comet J-P (2007) Necessary conditions for multistationarity in discrete dynamical systems. Discret Appl Math 155(18):2403–2413
Sackmann A, Heiner M, Koch I (2006) Application of Petri net based analysis techniques to signal transduction pathways. BMC Bioinformatics 7:482
Sánchez L, Chaouiya C, Thieffry D (2008) Segmenting the fly embryo: a logical analysis of the segment polarity cross-regulatory module. Int J Dev Biol 52(8):1059–1075
Schlitt T, Brazma A (2007) Current approaches to gene regulatory network modelling. BMC Bioinformatics 8:S9
Siebert H, Bockmayr A (2006) Incorporating time delays into the logical analysis of gene regulatory networks. Lect Notes Comput Sci 4210:169–183
Siebert H, Bockmayr A (2007) Context sensitivity in logical modeling with time delays. Lect Notes Comput Sci 4695:64–79
Simão E, Remy E, Thieffry D, Chaouiya C (2005) Qualitative modelling of regulated metabolic pathways: application to the tryptophan biosynthesis in *E. coli*. Bioinformatics 21:ii190–ii196
Soulé C (2006) Mathematical approaches to gene regulation and differentiation. CR Acad Sci Paris (Biol) 329:13–20
Srivastava R, Peterson MS, Bentley WE (2001) Stochastic kinetic analysis of the *Escherichia coli* stress circuit using $\sigma^3$2-targeted antisense. Biotechnol Bioeng 75:120–129
Steggles LJ, Banks R, Shaw O, Wipat A (2007) Qualitatively modelling and analysing genetic regulatory networks: a Petri net approach. Bioinformatics 23:336–343
Thieffry D (2007) Dynamical roles of biological regulatory circuits. Brief Bioinform 8:220–225
Thomas R (1991) Regulatory networks seen as asynchronous automata: a logical description. J Theor Biol 153(1):1–23
Thomas R, D'Ari R (1990) Biological feedback. CRC Press, Boca Raton
Thomas R, Thieffry D, Kaufman M (1995) 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(2):247–276