Decision Diagrams for the Representation and Analysis of Logical Models of Genetic Networks

Aurélien Naldi, Denis Thieffry, and Claudine Chaouiya

TAGC, INSERM ERM206, Université de la Méditerranée
Marseilles, France

Abstract. The complexity of biological regulatory networks calls for the development of proper mathematical methods to model their structures and to obtain insight in their dynamical behaviours. One qualitative approach consists in modelling regulatory networks in terms of logical equations (using either Boolean or multi-valued discretisation).

In this paper, we propose a novel implementation of the generalised logical formalism by means of Multi-valued Decision Diagrams. We show that the use of this representation enables the development of efficient algorithms for the analysis of specific dynamical properties of the regulatory graphs. In particular, we address the question of determining conditions insuring the functionality of feedback circuits, as well as the identification of stable states. Finally, we apply these algorithms to logical models of T cell activation and differentiation.

Keywords: Regulatory networks, logical modelling, decision diagrams, regulatory circuits, stable states.

1 Introduction

Modelling is a crucial step towards a functional understanding of the complex interaction networks that govern fundamental cellular processes. In this respect, in order to overcome the lack of quantitative data, logical approaches have been successfully applied to a wide variety of genetic networks involved in cell differentiation and pattern formation (for an introduction to logical modelling of genetic networks, see [12,3]). However, when facing very large regulatory networks, even logical abstraction leads to hard combinatorial problems. In other contexts, decision diagrams have been successfully applied to similar combinatorial problems, in particular for symbolic model-checking (*e.g.* [2]). Here, we show how decision diagrams can be used to represent sophisticated logical rules and enable the development of efficient algorithms to determine the conditions insuring specific dynamical roles for the different regulatory circuits, as well as to identify all the stable states of large and complex systems, without explicitly constructing the state transition graph and independently of any initial conditions.

Finally, we apply these algorithms to: (i) a model of Th1/2 differentiation [8] and (ii) a model of the TCR signalling pathway [7].

M. Calder and S. Gilmore (Eds.): CMSB 2007, LNBI 4695, pp. 233–247, 2007.

2 Logical Modelling of Gene Regulatory Networks

Our modelling approach leans on the generalised logical formalism initially developed by R. Thomas and collaborators [13,3]. In this context, a regulatory network and its dynamics are both represented in terms of oriented graphs.

2.1 Regulatory Graphs

A regulatory network is defined as a labeled directed graph $\mathcal{R} = (\mathcal{G}, \mathcal{A}, \mathcal{K})$ called *regulatory graph*, where:

- $\mathcal{G} = \{g_1, \ldots, g_n\}$ is the set of nodes of the regulatory graph, representing genes (or, more generally, regulatory components). Each $g_i \in \mathcal{G}$ is associated with its maximum *expression level* Max_i ($Max_i \in \mathbb{N}^*$) and its current expression level x_i ($x_i \in [0, Max_i]$).
- $\mathcal{A}$ is the set of arcs. An arc (g_i, g_j) specifies that the gene g_i regulates the gene g_j (when there is no possible confusion, we often write i for g_i). A regulatory graph may contain self-loops (*e.g.* a self-regulated gene g_i with an arc (i, i)).

  For each gene g_j, $Reg(j)$ denotes the set of its regulators: $i \in Reg(j)$ if and only if $(i, j) \in \mathcal{A}$.

  If $Max_i > 1$, g_i may have different effects onto a gene g_j, depending on the actual activity level of g_i. Thus the arc (i, j) may be indeed a multi-arc encompassing different *interactions*. The multiplicity of the arc (i, j) (*i.e.* the number of its constitutive interactions), is denoted $m_{i,j}$ ($1 \leq m_{i,j} \leq Max_i$). A *threshold* $\theta_{i,j,k}$ (integer taking its values in $[1, m_{i,j}]$) is associated to the k^{th} interaction (denoted $(i, j, k), k \in [1, m_{i,j}]$), with $1 \leq \theta_{i,j,1} < \ldots < \theta_{i,j,m_{i,j}} \leq Max_i$. The k^{th} interaction is *active*, when 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, $\theta_{i,j,m_{i,j}+1} = Max_i + 1$).
- $\mathcal{K} = (\mathcal{K}_1, \ldots, \mathcal{K}_n)$ defines the dynamics of the system: each $\mathcal{K}_i$ is a multi-valued logical function defining the evolution of the variable x_i, depending on the incoming active interactions of g_i:

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

  For example, if g_2 and g_3 are regulators of g_1 ($Reg(1) = \{2, 3\}$), $\mathcal{K}_1(0, 1)$ is the focal value of g_1 when no interaction from g_2 is active (*i.e.* $x_2 < \theta_{2,1,1}$) and the first interaction from g_3 is active ($\theta_{3,1,1} \leq x_3 < \theta_{3,1,2}$).

Note that the biologists often consider *different types* of interactions: activations (*resp.* repressions) have a positive (*resp.* negative) effect on their targets. However the actual effect of an interaction often depends 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.

2.2 Dynamics of a Regulatory Graph

A state x of the regulatory graph is a n-tuple $(x_1, \ldots, x_n)$ of the expression levels of the genes: $x \in \prod_{i=0}^{n}[0, Max_i]$. Given a state and for each gene g_i, it is then possible to determine the set of interactions operating onto g_i (the active interactions). We thus define the functions $\mathcal{K}'_i(x)$s, which follow from the logical functions $\mathcal{K}_i$s and directly depend on the current state x of the system:

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

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

In the following, for simplicity, $\mathcal{K}'_i$ will be denoted $\mathcal{K}_i$ (omitting the prime sign).

Given the current state x, the level of each g_i tends toward the *focal 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.

The dynamics of the regulatory graph can then be represented by a *state transition graph*, where nodes represent states (giving the levels of the regulatory components) and arcs represent transitions between states (*i.e.* changes of the values of some components). This state transition graph is computed by means of the $\mathcal{K}_i$s, which indicate the transitions leading from the current state to its following states (here, we consider an asynchronous updating, where each transition corresponds to a change of a unique variable, see [3] for further details). When facing large regulatory networks (*i.e.* dozens or hundreds of components), combinatorial problems impede the full computation of state transition graphs. In this paper, we assess the use of Decision Diagrams to handle this combinatorial problem for complex multi-valued logical models.

3 Regulatory Graphs and Multi-valued Decision Diagrams

Multi-valued logical functions have a finite number of possible values, depending on a set of *decision variables*, which also take a finite range of values. Such functions can be represented using efficient data structures (see [1] for further details).

Decision Diagrams are particularly promising in our context. Indeed, Garg et al. have already used BDD to represent the whole state transition graph of Boolean models of biological regulatory networks (their approach is contrasted with ours in the conclusion).

In the following section, we recall the definitions at the basis of our novel implementation of logical functions.

3.1 Decision Diagrams

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ can be represented as a *binary decision tree* where non-terminal nodes are labelled by a decision variable and where terminal nodes are labelled either 1 (true) or 0 (false). The edge from a decision node to its left child (*resp.* right child) corresponds to an assignment of the variable to 0 (*resp.* to 1). Given a state $x \in \{0,1\}^n$ (defining the values of n decision variables), a unique path from the root to a terminal node (a leaf) is defined. Along this path, the child chosen for each non-terminal node is labelled with the value of the corresponding variable in state x. The terminal node reached through this path gives the value of $f(x)$.

Reduced Binary Decision Diagrams (RBDDs) have been introduced to improve this representation, which requires exponential space ($2^{n+1} - 1$ nodes). RBDDs are obtained applying two reduction rules: (i) merge isomorphic subgraphs and (ii) bypass nodes whose children are the roots of isomorphic subdiagrams. The resulting structure is then a rooted directed acyclic graph. Bryant further extended this representation by the use of a fixed variable ordering which leads to canonical representations of logical functions. The resulting graphs are called Reduced *Ordered* BDDs (ROBDDs), commonly referred to as BDDs. Note that the size of BDDs may depend on the variable ordering (see Figure 1 for an illustration of the impact of the ordering).

BDDs have been generalised to the multi-valued case: a discrete multi-valued function can be represented by a *Multi-valued Decision Diagram* (MDD), where decision nodes may have as many children as the number of their possible values and the terminal nodes are labelled by the values of the function (see Figure 1 for an illustration) [6]. The ordering and reduction rules defined for BDDs apply also to this multi-valued generalisation.

3.2 Use of MDDs to Represent Logical Functions

The functions $\mathcal{K}_i$s, which take their values in $[0, Max_i]$, can be represented as MDDs, with the regulators of g_i as decision variables. During the simulation (*i.e.* the construction of the state transition graph), the focal value of a gene g_i is obtained in $O(\#Reg(i))$ in the worst case, traversing the corresponding MDD. This data structure thus definitely improved the performance of GINsim, our software implementing the logical formalism [5].

Moreover the representation of the $\mathcal{K}_i$s by means of MDDs greatly facilitates the analysis of specific dynamical properties as showed in the following sections.

In the sequel, for simplicity, diagrams will be named after the multi-valued functions they represent. For a given regulatory graph, an arbitrary ordering on the set of variables is used consistently for all the related MDDs.

4 Analysis of Regulatory Circuits

In what follows, we consider *elementary circuits* (circuits, for short), *i.e.* finite closed paths in the regulatory graph, where all the nodes are distinct.

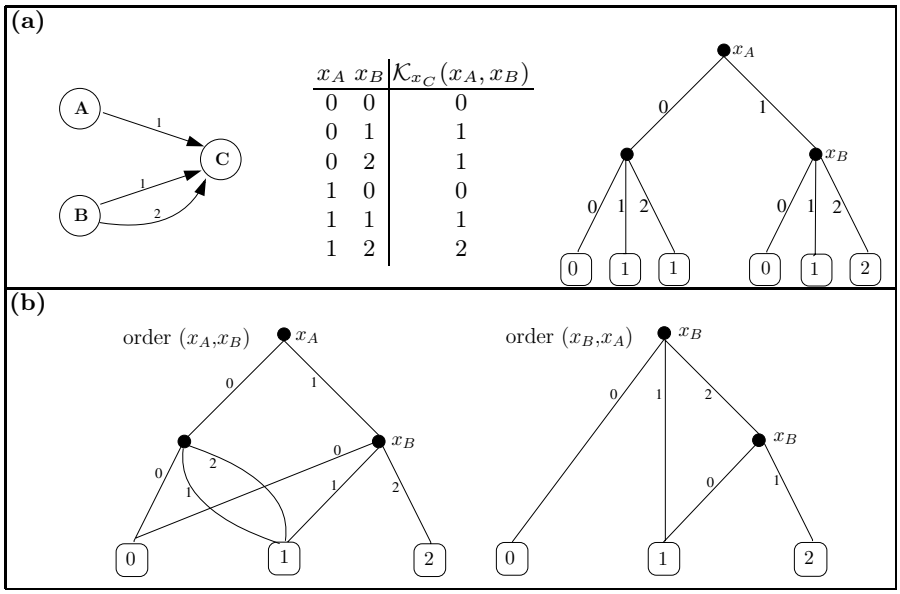
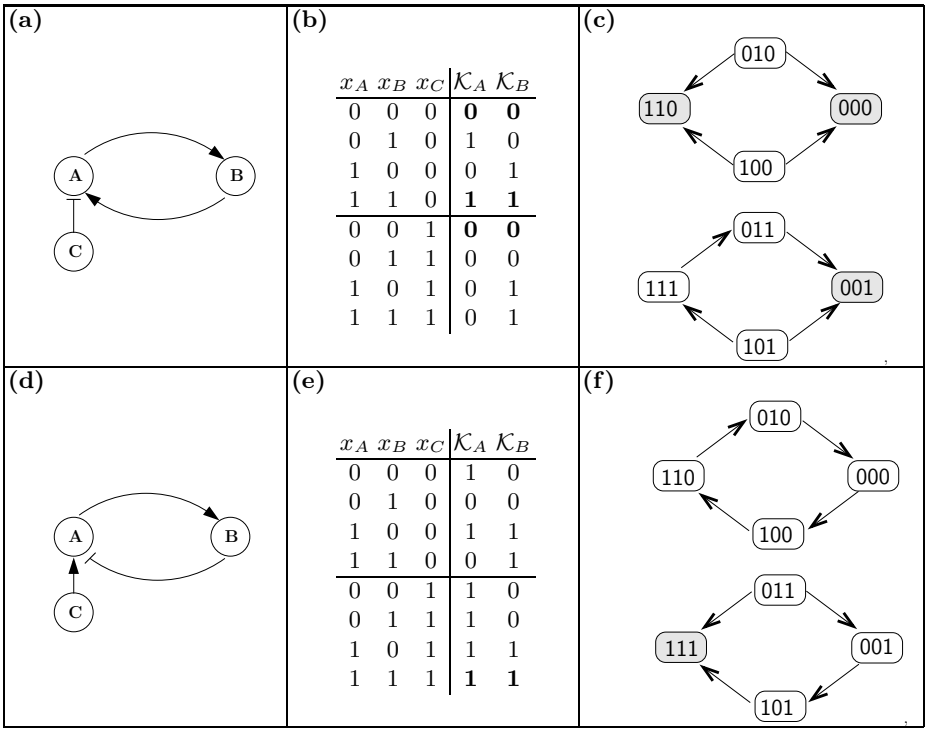


Fig. 1. Example of a simple logical regulatory graph with its MDD representation: **(a)** The regulatory graph, the table defining the function $\mathcal{K}_C$ together with its decision tree representation. **(b)** The reduced MDDs (considering the two different ordering of x_A and x_B) representing $\mathcal{K}_C$, with x_A and x_B as decision variables and the values of $\mathcal{K}_C$ labelling the leaves.

It is well established that complex dynamical behaviours of regulatory networks are related to their topological structures. In particular, the roles of regulatory circuits have been emphasised by R. Thomas, who proposed that negative circuits are required to observe oscillatory behaviours, whereas positive circuits are necessary for multistationarity (the sign of a circuit is given by the product of the signs of its interactions) [11]. Proofs of these conditions have been presented within different modelling formalisms [10,9]. But the sole presence of a circuit in a network does not guarantee the appearance of the corresponding dynamical behaviour. The circuit must be *functional* [11]. Figure 2 illustrates how the regulators of one of its components can prevent a circuit from generating the expected behaviour. We thus define the *context of functionality* of a circuit as a set of constraints on the values of the external inputs acting on that circuit. Our definition of functionality context may serve as a basis to formally prove the relationship between the functionality of a circuit and the corresponding dynamical properties.

In the multi-valued case, circuits containing multiple interactions can be splitted into multiple "elementary circuits" and considered separately. In the sequel we restrict ourselves to the case of single interactions to simplify the notation: the threshold of an interaction can be thus denoted $\theta_{i,j}$ (instead of $\theta_{i,j,k}$).



x_A	x_B	x_C	$\mathcal{K}_A$	$\mathcal{K}_B$
0	0	0	**0**	**0**
0	1	0	1	0
1	0	0	0	1
1	1	0	**1**	**1**
0	0	1	**0**	**0**
0	1	1	0	0
1	0	1	0	1
1	1	1	0	1

x_A	x_B	x_C	$\mathcal{K}_A$	$\mathcal{K}_B$
0	0	0	1	0
0	1	0	0	0
1	0	0	1	1
1	1	0	0	1
0	0	1	1	0
0	1	1	1	0
1	0	1	1	1
1	1	1	**1**	**1**

Fig. 2. Illustration of the influence of external inputs on the dynamics of a regulatory circuit. In the regulatory graphs of panels **(a)** and **(d)**, activations are depicted by normal arrows whereas inhibitions are depicted by blunt arrows. **(a)** A simple positive circuit affected by a negative input. **(b)** Values for $\mathcal{K}_A$ and $\mathcal{K}_B$: two situations arise, depending on the value of x_C (component A regulated by both B and C). **(c)** The two possible behaviours, depending on x_C: for $x_C = 0$ there are two stable states (top), while a unique stable state exists for $x_C = 1$ (bottom). **(d,e,f)** A simple negative circuit affected by a positive input; oscillations only appear in the absence of the input.

4.1 Functionality Context and Sign of an Interaction

In general, we say that an interaction (i, j) is functional when it affects the focal value of its target: $\mathcal{K}_j(x_1, \ldots, \theta_{i,j} - 1, \ldots, x_n) \neq \mathcal{K}_j(x_1, \ldots, \theta_{i,j}, \ldots, x_n)$. In the context of a circuit, this change must further affect the activity of the next interaction in the circuit (the threshold of the next interaction must be crossed to reach the focal value). This additional constraint is only relevant for multi-valued genes, as it is always satisfied in the Boolean case.

In the following, we define the functionality of the interaction (i, j) and its context, that is the set of constraints upon the regulators of j (target of the considered interaction).

Definition 1. *Let consider an interaction (i, j), member of a circuit $\mathcal{C}$ with a threshold $\theta_{i,j}$. Let (j, k) be the next interaction in $\mathcal{C}$, with a threshold $\theta_{j,k}$. The*

interaction (i, j) is said to be functional in $\mathcal{C}$ if and only if there exists a variable assignment for all regulators of g_j except g_i such that:

$$\mathcal{K}_j(x_1, \ldots, x_{i-1}, \theta_{i,j} - 1, \ldots, x_n) < \theta_{j,k} \leq \mathcal{K}_j(x_1, \ldots, x_{i-1}, \theta_{i,j}, \ldots, x_n), \quad (1)$$

$$\text{or} \quad \mathcal{K}_j(x_1, \ldots, x_{i-1}, \theta_{i,j}, \ldots, x_n) < \theta_{j,k} \leq \mathcal{K}_j(x_1, \ldots, x_{i-1}, \theta_{i,j} - 1, \ldots, x_n). \quad (2)$$

The functionality context of the interaction (i, j) in $\mathcal{C}$ is defined as the subset of $\Pi_{k=1}^{n}[0, Max_k]$ of n-tuples such that the values of the regulators of g_j let the interaction (i, j) functional (i.e. Equation (1) or (2) satisfied). The interaction is thus functional if its context is not empty.

Definition 1 establishes that the interaction (i, j) is functional provided its activity affects the activity of the following interaction of the circuit (going out g_j). This depends on the values of $\mathcal{K}_j$, considering values $\theta_{i,j} - 1$ and $\theta_{i,j}$ for g_i and all possible values of other regulators of g_j.

We can then define the sign of an interaction, as 0 when it is not functional, or 1 (*resp.* -1) when it is functional and leads to an increase (*resp.* a decrease) of the focal value of its target.

Definition 2. *Let us consider consecutive interactions (i, j) and (j, k) in a circuit $\mathcal{C}$. Given a variable assignment for all regulators of g_j (except g_i), the sign of (i, j) in $\mathcal{C}$ is given by $\Gamma_{i,j}$, defined as follows:*

$$\Gamma_{i,j}(x) = \begin{cases} 1 & \text{if Equation (1) holds,} \\ -1 & \text{if Equation (2) holds,} \\ 0 & \text{otherwise,} \end{cases}$$

where x is a state ($x \in \Pi_{k=1}^{n}[0, Max_k]$), for which the values corresponding to the regulators of g_j (except that of g_i) equal the given assignment.

We say that (i, j) has a positive effect when $\Gamma_{i,j}(x) = 1$, a negative effect when $\Gamma_{i,j}(x) = -1$ and no effect otherwise.

The construction of the MDD representing $\Gamma_{i,j}$ for a given interaction (i, j) is illustrated in Figure 3**(a-b)**. The algorithm is given as supplementary material. The leaves of the MDD give the sign of (i, j), depending on the path followed to reach them. This path defines the conditions on the values taken by the regulators of g_j.

Remark 1. It may happen that the sign of an interaction (i, j) is *context dependent*, that is, for an assignment of the regulators of g_j (except g_i) the sign of the interaction is positive and for another assignment, it is negative. To simplify the explanations in the next section (that defines the functionality of a whole circuit), we exclude such a case, which is infrequent in genetic regulatory networks.

4.2 Functionality Context and Sign of a Regulatory Circuit

We now consider the case of a whole elementary circuit. Definition 3 formalises the functionality context of a circuit, as well as its sign, depending on its constitutive interactions.

First, we will consider the case of circuits which do not contain smaller circuit(s), or shortcuts. A circuit $\mathcal{C} = (c_1, c_2, \ldots, c_r)$ (with $r + 1 = 1$) contains a shortcut if there exists c_i which regulates c_k, with $k \neq i + 1$ ($c_i \in \mathcal{C}$ and $c_k \in \mathcal{C}$). The simplest example of such a shortcut is an auto-regulation of a member of the circuit. Then, we extend the definition of the functionality to the general case (with, for simplicity, the restriction that no interaction of the circuit has a context dependent sign).

Definition 3. *The functionality context of a circuit with no shortcut is defined as the intersection of the functionality contexts of its constitutive interactions; the circuit is thus functional when this intersection is not empty (implying that all its interactions are functional). The sign of the circuit $\mathcal{C} = (c_1, c_2, \ldots, c_r)$ is defined as the product of the signs of its interactions:*

$$\Gamma_{\mathcal{C}}(x) = \prod_{i=1}^{i \leq r} \Gamma_{c_i, c_{i+1}}(x).$$

Definition 4. *Let consider a circuit $\mathcal{C} = (c_1, c_2, \ldots, c_r)$ which contains shortcut(s). Its functionality context $\Gamma_{\mathcal{C}} \subseteq \Pi_{k=1}^{n}[0, Max_k]$ is defined as the intersection of the contexts of its interactions further restricted to insure that, for all c_i acting on a component c_k of $\mathcal{C}$ different from c_{i+1}, and an assignment $\bar{y}$ of all variables but c_i:*

$$(y_1, \ldots, \theta_{i,i+1} - 1, \ldots y_n) \in \Gamma_{\mathcal{C}} \Longleftrightarrow (y_1, \ldots, \theta_{i,i+1}, \ldots y_n) \in \Gamma_{\mathcal{C}}.$$

When the above condition does not hold, all the tuples $(y_1, \ldots, x_i, \ldots y_n)$ are removed from $\Gamma_{\mathcal{C}}$ (for all possible values x_i of c_i). The circuit $\mathcal{C}$ is functional if its functionality context $\Gamma_{\mathcal{C}}$ is not empty. Its sign is defined as the product of the signs of its interactions.

The above definition guarantees that, given a fixed assignment of all other variables compatible with $\Gamma_{\mathcal{C}}$ (the functionality context of $\mathcal{C}$), both states where c_i level is less or equal to $\theta_{i,i+1}$ (the threshold of the interaction of $\mathcal{C}$ going from c_i to c_{i+1}) are in $\Gamma_{\mathcal{C}}$ (and this insures the functionality of this interaction).

The determination of the function Γ_C requires two operations: (i) the determination of all the $\Gamma_{i,j}$, giving the signs of single interactions; and (ii) the computation of their product.

Assume that the diagrams giving the signs of the interactions composing the circuit have been determined. The MDD giving the global sign of the circuit is built as a product of the signs of the individual interactions. This product is performed by means of the combination of MDDs Γ_1 and Γ_2, applying the following rules (see Figure 3**(c)** for an illustration):

- if $\Gamma_1$and Γ_2 are reduced to single nodes, the product is a node, its value being the product of those of $\Gamma_1$and Γ_2;
- else if Γ_1 (resp Γ_2) is a single node with value 1, the result is Γ_2 (*resp.* Γ_1);

- else if Γ_1 and Γ_2 roots are internal nodes with the same order (hence corresponding to the same decision variable), the result is a root of this order, and its children are recursive combinations of those of Γ_1 and Γ_2 roots;
- otherwise, if Γ_2 is a single node with value (-1) or if the root of Γ_1 has an order less than that of the root of Γ_2 (or symmetrically), the result is a root such that:
  - its order is the order of the root of Γ_1;
  - its children are recursive combinations of Γ_2 with those of Γ_1.

Finally, in the case of shortcuts in the circuit, as the sign of the circuit $\mathcal{C}$ may depend on the levels of members of $\mathcal{C}$, this dependancy is properly removed while ensuring that every member of the circuit can cross its threshold. Further details can be found in the supplementary material.

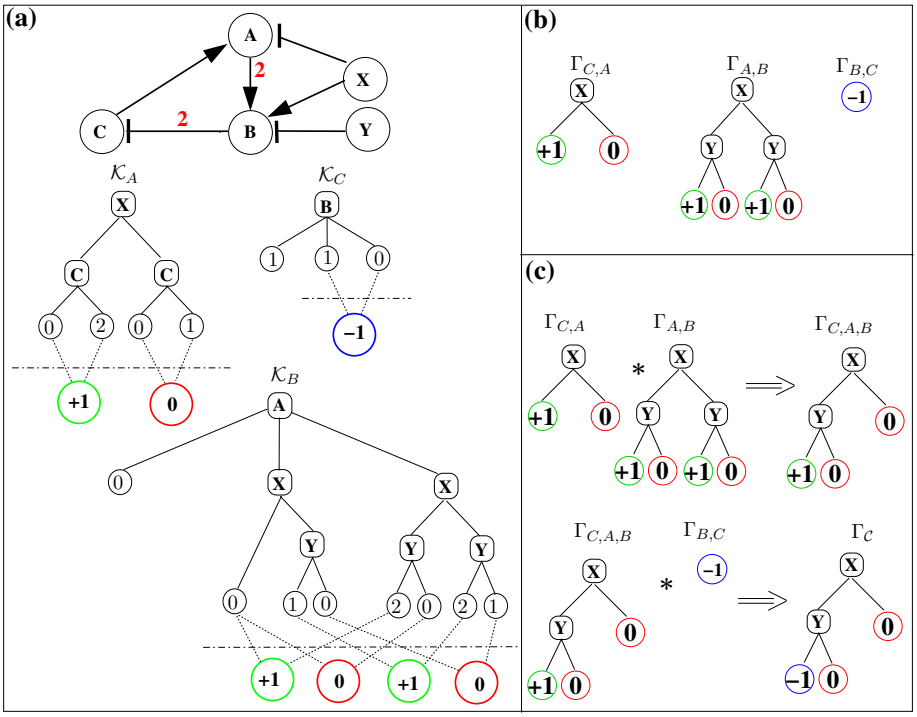


Fig. 3. Determination of the functionality context (and sign) of a circuit: **(a)** A regulatory network encompassing a circuit and decision diagrams of the logical functions $\mathcal{K}$s for the three members of the circuit. Below the logical functions $\mathcal{K}$s, pairwise comparisons of the leaves delineate the signs of the interactions targeting the members of the circuit. **(b)** MDDs giving the signs Γs of the interactions. **(c)** Combinations of the MDDs to determine $\Gamma_{\mathcal{C}}$, the sign of the circuit. The paths in $\Gamma_{\mathcal{C}}$ leading to non-zero leaves give the functionality context of the circuit. For sake of clarity, the diagrams are not fully reduced.

5 Efficient Determination of Logical Stable States

In this section, we show how the MDD representation of the logical functions $\mathcal{K}_i$s can be used to determine all the logical stable states of a parameterised regulatory graph. A stable state x is such that the focal value of each gene is identical to its current value:

$$x \in \Pi_{i=1}^{n}[0, Max_i] \text{ is stable } \textit{iff } \mathcal{K}_i(x) = x_i, \ \forall i \in \{1, \ldots n\}. \tag{3}$$

The algorithm to determine the stable states encompasses two main steps. First, for each gene g_i, a MDD $\mathcal{S}_i$ is constructed, which gives the logical stability condition depending on its value x_i and on those of its regulators (box (b) in Figure 4). Second, the resulting MDDs are combined as described in 4.2.

For a gene g_i, the first step amounts to transform the MDD representing the logical function $\mathcal{K}_i$. The decision variable x_i is properly added and the leaves values are set to 0 for a change (a decrease or an increase), or to 1 for no change. The resulting MDD implements a logical function with value 1 (true) when the node is stable, 0 (false) otherwise:

$$\mathcal{S}_i : \prod_{j \in Reg(i)} [0, Max_j] \rightarrow \{0, 1\},$$

with

$$\mathcal{S}_i = \begin{cases} 0 \text{ if } \mathcal{K}_i(x) \neq x_i, \\ 1 \text{ if } \mathcal{K}_i(x) = x_i. \end{cases}$$

To simplify the notation, $S_i(x)$ and $K_i(x)$ are considered beyond the sole regulators of i, to encompass all components of x. The diagrams $\mathcal{S}_i$ are then pairwise combined (the combination of $\mathcal{S}_i$ and $\mathcal{S}_j$ is the representation of the logical stability condition for both g_i and g_j). As for the determination of the sign of a circuit, each combination amounts to the product of the corresponding MDDs. Ultimately, the diagram $\mathcal{S}_{1\ldots n}$ defines the stability condition for the whole set of genes, hence the paths leading to 1-leaves give the stable states (see Figure 4).

6 Application to Regulatory Networks Controlling T Cell Activation and Differentiation

We have applied our novel analysis tools to two logical models recently published: (i) the first (multi-valued) model accounts for the differentiation of naive T-helper lymphocytes into two subtypes, called Th1 and Th2, controlling cellular and humoral immune responses, respectively [8]; (ii) the second, a Boolean model, integrates the information available on T cell receptor (TCR) signalling, taking into account several co-acting signals [7]. These two models are closely related, as NFAT, one of the outputs of the TCR signalling pathway, is one of the activators of the IFN-γ pathway.

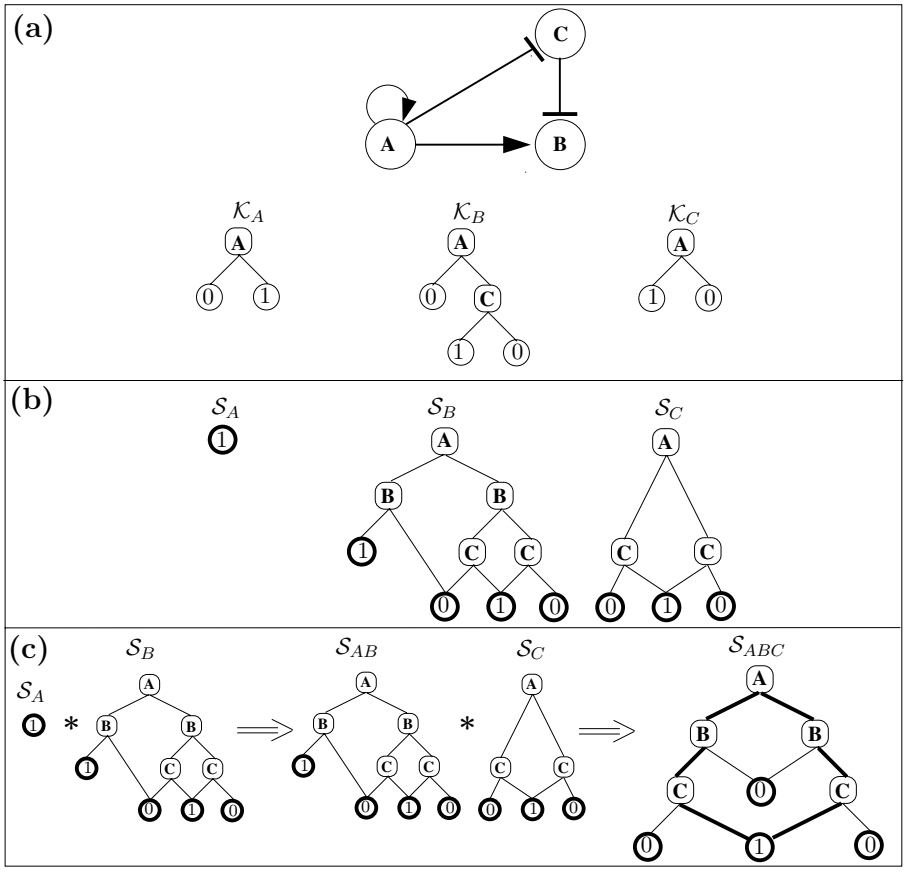


Fig. 4. Illustration of the stable state determination: **(a)** a simple regulatory graph and the associated logical functions; **(b)** the diagrams $\mathcal{S}$ giving the constraints for each gene to be stable; **(c)** the combination of the diagrams to obtain $\mathcal{S}_{ABC}$. For $\mathcal{S}_{ABC}$, two paths lead to the 1-leaf (they are depicted in bold), indicating that the regulatory graph has two stable states: $(x_A, x_B, x_C) = (0, 0, 1)$ and $(1, 1, 0)$. Note that, for clearness, the diagrams are not fully reduced.

6.1 T Cell Differentiation

The regulatory graph is shown in Figure 5. The graph encompasses 17 regulatory components, including five cytokines or intercellular signalling molecules, the interferons beta and gamma, and the interleukines 4, 12 and 18, the corresponding receptors, five mediatory molecules, SOCS1, IRAK, and STAT1, 4 and 6, as well as two transcription factors, Tbet and GATA3. All components but four are modelled by Boolean variables. The cascade involving IFN-γ, its receptor, STAT1 and Tbet are modelled by ternary variables as cells presenting two different levels of activation of the IFN-γ pathway have been

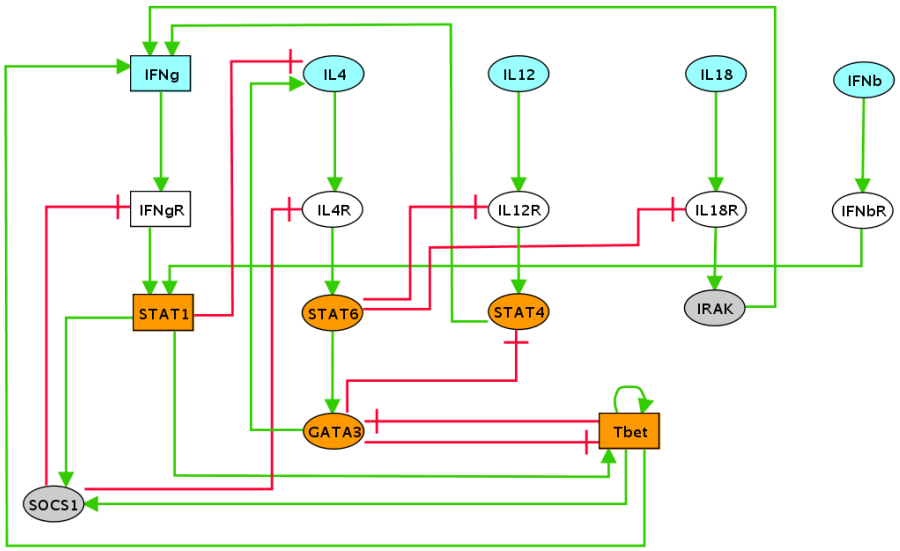


Fig. 5. The network controlling Th1/2 differentiation. All regulatory components but four are modelled by Boolean variables. The remaining four (rectangular nodes) are modelled by ternary variables. The first two nodes layers denote cytokines and their receptors. Normal arrows represent activations, whereas blunt arrows represent inhibitory effects. Note that the original model of Mendoza encompasses two variants, including an auto-activation for GATA3 (murine cells) or not (human cells).

experimentally observed. The experimental information in support of this graph, as well as the delineation of the logical parameters can be found in the original article published by Mendoza [8]. The logical model can be downloaded in a GINsim dedicated XML format from the address http://gin.univ-mrs.fr/GINsim/.

The algorithm presented in Section 5 takes less than a second to identify the four stable states reported by Mendoza:

- a *Th0* state without any active component, corresponding to the naive, undifferentiated cell;
- a *Th1* state where IFN-γ, IFN-γR, STAT1, SOCS1 and Tbet are expressed;
- a *Th1** state, similar to the previous one, but with higher expression levels for IFN-γ, and T-bet (the expression of SOCS1 prevents IFN-γR and STAT1 from showing a higher expression level);
- a *Th2* state showing expression of IL4, IL4R, STAT6 and GATA3.

Turning to the circuit functionality analysis, the algorithm described in Section 4 leads to the identification of five functional circuits among the 22 circuits found in the regulatory graph for the human model variant shown in Figure 5. All of these functional circuits are positive, which is consistent with the fact that this network is predominantly involved in the control of cell differentiation.

- The auto-regulation of T-bet is functional in the absence of both GATA3 and STAT1, or yet in the presence of a medium level of STAT1. Note that this circuit involves two different levels of Tbet and can thus enable the presence of up to three stable states, each characterized by one of the possible expression levels of Tbet.
- The (GATA3, IL4, IL4R, STAT6) circuit is functional in the absence of STAT1, SOCS1 and T-bet. This circuit ensures a coupling between the expressions of GATA3, IL4, IL4R and STAT6. In the murine cells, this circuit is not functional, replaced by the GATA3 auto-activation (functional in the absence of Tbet and STAT6).
- The (GATA3, T-bet) circuit is functional in the presence of STAT1 and STAT6. This cross-inhibitory circuit ensures the exclusive expressions of the transcriptional regulators Tbet and GATA3, characteristic of Th1 and Th2 responses, respectively. In the absence of the activators of Tbet and GATA3 (STAT1 and STAT6), the cell remains trapped in the naive state.
- The last two functional circuits involve IFN-γ, IFN-γR, STAT1, SOCS1, STAT6, and STAT4, plus a few additional components. Their contexts of functionality are relatively restrictive (absence of Tbet and IFNbR, and presence of IL12 and IL4, plus the absence of GATA3 in one of the two cases).

On the basis of the results of the circuit functionality analysis, it is possible to delineate specific perturbations affecting the stable state configuration.

6.2 T Cell Activation

The model recently published by Klamt et al. for T cell activation encompasses 40 regulatory components, hierarchically organised, from cell membrane receptors (TCR, CD8 and CD45) to transcription factors (CRE, AP1, NF-κB, NFAT) [7]. After adding three functional auto-activations on the input nodes (TCR, CD8 and CD45), our stable state identification tool identifies seven alternative stable states, each corresponding to a specific input configuration. Strikingly, the input configuration with all receptors permanently activated does not lead to any stable state, but rather to a complex periodic behaviour. However, in normal physiological situations, one should expect only transient activations of these receptors, thus ultimately leading to a unique stable state, corresponding to the resting situation.

In this respect, cross-talks between the signalling cascades may play an important role. Here, our circuit functionality analysis tool can be useful. Apart from the auto-activations purposively added on each of the three receptors, its application to this system leads to the identification of only one negative functional circuit out of nine: (ZAP70, cCBL). Under full stimulation (*i.e.* in the presence of all three inputs), this circuit enables an oscillatory behaviour of the involved regulatory components. These oscillations propagate downstream, leading to cyclic expression of the transcription factors (outputs of this model). In physiological situations, these oscillations must abort following the natural receptor inactivation.

7 Conclusion

In this paper, we have shown how Multi-valued Decision Diagrams (MDDs) can be used to encode the logical functions governing the behaviours of individual nodes in qualitative models of complex regulatory networks. Leaning on this encoding, we have further delineated two algorithms, one to efficiently determine all stable states of a logical model, the other to compute the functionality context of each regulatory circuit, in terms of conditions on the values of the variables acting on this circuit.

As mentionned above, Garg *et al.* have already represented Boolean state transition graphs in terms of BDD. They considered the particular case of networks where genes are expressed provided all their inhibitors are absent and at least one of their activators are present [4]. Based on their BDD representation, the authors propose an efficient method to determine stable states, and even more complex attractors. In contrast, our proposal refer to multi-valued logical networks where the logical functions can be more subtle. More importantly, our approach is based on a Decision Diagram representation of the transition functions for each node. Stable states, as well as feedback circuit functionality, are then determined through proper combinations of these diagrams.

On the basis of a prototype implementation of these algorithms, we are presently analysing a series of regulatory models (Boolean or multi-valued) involved in cell differentiation and pattern formation, encompassing dozens of regulatory components, involved in hundreds of regulatory circuits. For each of these systems, the delineation of all stable states and the computation of feedback circuit functionality domains took less than a second on a standard computer.

In section 6, we have briefly presented the results obtained for two recently published logical models: (i) a multi-valued model of the network controlling T-helper lymphocytes differentiation; (ii) a Boolean model encompassing some of the main signalling cascades controlling the activation of T cells.

At this point, we believe that it should be possible to further improve the performance of our algorithms. In particular, a proper ordering of decision variables can have a significant impact on the overall sizes of the MDDs (note that a common ordering of the variables must be defined to ensure a coherent combination of the MDDs). Similarly, we observed that the order of consecutive MDD combinations (*e.g.* in the course of the identification of all stable states) has a strong effect on the overall performance.

Finally, this MDD representation opens interesting prospects for the modelling of combinations of mutations or other perturbations through a rewriting of the MDDs describing the wild-type model.

Supplementary Material

Further details on the algorithms, as well as on the T-helper cell differentiation and activation models are available at the following URL:
http://gin.univ-mrs.fr/GINsim/publications/naldi2007.html.

Acknowledgements

We acknowledge financial support from the European Commission (contract LSHG-CT-2004-512143), the French Research Ministry through the ANR project JC05-53969 and A. Naldi PhD grant.

References

1. Bryant, R.E.: Graph-based algorithms for boolean function manipulation. IEEE Trans. Comput. 35, 677–691 (1986)
2. Burch, J.R., Clarke, E.M., Long, D.E., MacMillan, K.L., Dill, D.L.: Symbolic Model Checking for Sequential Circuit Verification. IEEE Trans. Comput.-Aided Design Integrated Circuits 13, 401–424 (1994)
3. Chaouiya, C., Remy, E., Mossé, B., Thieffry, D.: Qualitative analysis of regulatory graphs: a computational tool based on a discrete formal framework. Lect. Notes Comp. Inf. Sci. 294, 119–126 (2003)
4. Garg, A., Xenarios, I., Mendoza, L., DeMicheli, G.: An Efficient Method for Dynamic Analysis of Gene Regulatory Networks and in-silico Gene Perturbation Experiments. Lect. Notes Comp. Sci. 4453, 62–76 (2007)
5. González, A., Naldi, A., Sanchez, L., Thieffry, D., Chaouiya, C.: GINsim: A software suite for the qualitative modelling, simulation and analysis of regulatory networks. Biosystems 84, 91–100 (2006)
6. Kam, T., Villa, T., Brayton, R.K., Sangiovanni-Vincentelli, A.L.: Multi-valued decision diagrams: Theory and applications. Int. J. Multiple-Valued Logic 4, 9–12 (1998)
7. Klamt, S., Saez-Rodriguez, J., Lindquist, J.A., Simeoni, L., Gilles, E.D.: A methodology for the structural and functional analysis of signaling and regulatory networks. BMC Bioinformatics 7(56) (2006)
8. Mendoza, L.: A network model for the control of the differentiation process in Th cells. Biosystems 84, 101–114 (2006)
9. Remy, E., Ruet, P., Mendoza, L., Thieffry, D., Chaouiya, C.: From logical regulatory graphs to standard petri nets: Dynamical roles and functionality of feedback circuits. Lect. Notes Comp. Sci. 4230, 55–72 (2006)
10. Soulé, C: Graphic requirements for multistationarity. ComPlexUs 1, 123–133 (2003)
11. Thomas, R.: On the relation between the logical structure of systems and their ability to generate multiple steady states of sustained oscillations. Springer Series Synergetics 9, 180–193 (1988)
12. Thomas, R.: Regulatory networks seen as asynchronous automata: A logical description. J. Theor. Biol. 153, 1–23 (1991)
13. Thomas, R., Thieffry, D., Kaufman, M.: 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, 247–276 (1995)