Dynamical modeling and analysis of large cellular regulatory networks

Cite as: Chaos **23**, 025114 (2013); https://doi.org/10.1063/1.4809783
Submitted: 16 January 2013 . Accepted: 08 May 2013 . Published Online: 25 June 2013

D. Bérenguier, C. Chaouiya, P. T. Monteiro, A. Naldi, E. Remy, D. Thieffry, and L. Tichit

Dynamical modeling and analysis of large cellular regulatory networks

D. Bérenguier,[1] C. Chaouiya,[2,a)] P. T. Monteiro,[2,3] A. Naldi,[4] E. Remy,[1,b)] D. Thieffry,[5,c)] and L. Tichit[1,6]
[1]*Institut de Mathématiques de Luminy, Marseille, France*
[2]*Instituto Gulbenkian de Ciência, Oeiras, Portugal*
[3]*Instituto de Engenharia de Sistemas e Computadores—Investigação e Desenvolvimento (INESC-ID), Lisboa, Portugal*
[4]*Center for Integrative Genomics, University of Lausanne, Lausanne, Switzerland*
[5]*Institut de Biologie de l'Ecole Normale Supérieure, Paris, France*
[6]*Aix-Marseille University, Marseille, France*

(Received 16 January 2013; accepted 8 May 2013; published online 25 June 2013)

The dynamical analysis of large biological regulatory networks requires the development of scalable methods for mathematical modeling. Following the approach initially introduced by Thomas, we formalize the interactions between the components of a network in terms of discrete variables, functions, and parameters. Model simulations result in directed graphs, called state transition graphs. We are particularly interested in reachability properties and asymptotic behaviors, which correspond to terminal strongly connected components (or "attractors") in the state transition graph. A well-known problem is the exponential increase of the size of state transition graphs with the number of network components, in particular when using the biologically realistic asynchronous updating assumption. To address this problem, we have developed several complementary methods enabling the analysis of the behavior of large and complex logical models: (i) the definition of transition priority classes to simplify the dynamics; (ii) a model reduction method preserving essential dynamical properties, (iii) a novel algorithm to compact state transition graphs and directly generate compressed representations, emphasizing relevant transient and asymptotic dynamical properties. The power of an approach combining these different methods is demonstrated by applying them to a recent multilevel logical model for the network controlling CD4+ T helper cell response to antigen presentation and to a dozen cytokines. This model accounts for the differentiation of canonical Th1 and Th2 lymphocytes, as well as of inflammatory Th17 and regulatory T cells, along with many hybrid subtypes. All these methods have been implemented into the software GINsim, which enables the definition, the analysis, and the simulation of logical regulatory graphs. [http://dx.doi.org/10.1063/1.4809783]

The dynamical analysis of comprehensive biological regulatory networks requires the development of scalable mathematical modeling methods. In this context, discrete (Boolean or multilevel) logical modeling is increasingly used to handle and analyze large molecular networks.[18] This article focuses on the presentation of several approaches to cope with the inherent exponential growth of the discrete state space as the size of the regulatory networks considered increases.

I. INTRODUCTION

To model biological regulatory networks, we rely on the logical approach initially introduced by Thomas, where the interactions between the components of a network are formalized in terms of discrete variables, functions, and parameters.[10,37,38] This modeling approach has proved effective in its application to a variety of regulatory and signaling networks, from yeast cell cycle control[15] to T lymphocyte differentiation.[27]

The logical modeling approach has been implemented into the software GINsim, which enables the definition of *logical regulatory graphs* and provides a number of original functionalities. These include the construction of synchronous or asynchronous *state transition graphs (STGs)* that represent model dynamical behaviors, along with algorithms enabling the determination of all logical stable states and the analysis of the roles of regulatory circuits.[9,26] However, when focusing on transient aspects of the dynamics or on the reachability of the attractors from specific initial conditions, we are facing the recurrent combinatorial explosion inherent in these models: the size of the state space grows exponentially with the number of regulatory components involved in the model. This problem is particularly acute in the case of asynchronous, non-deterministic updating mode, which is usually more biologically realistic than the simpler, deterministic synchronous updating mode.[17,37] Here, we present

a)Electronic mail: chaouiya@igc.gulbenkian.pt
b)Electronic mail: elisabeth.remy@univ-amu.fr
c)Electronic mail: thieffry@ens.fr

an overview of three main complementary strategies to cope with this combinatorial explosion.

The first strategy consists in reducing the model before performing simulations or other kinds of analysis.[29]

The second strategy simplifies the state transition graphs by forcing choices between alternative transitions; this can be achieved by defining priority (a/synchronous) transition classes, which are similar to time-scale based assumptions often used to simplify the dynamical analysis of ordinary differential equation (ODE) models.[14]

The third, novel strategy consists in compressing the state transition graph into a novel graph representation, called *hierarchical transition graph (HTG)*, which keeps track of attractors and their basins of attraction, as well as of transient oscillatory properties; here, we further propose an algorithm for the construction of this hierarchical graph. We also show that this method can be used in combination with the two aforementioned approaches to get insights into the dynamics of complex logical regulatory graphs.

In addition, model-checking approaches rely on symbolic representations of the dynamics, exploring only the necessary state space required for the verification of properties expressed as temporal logic formulas.

Section II introduces the basics of the multilevel logical formalism and provides an overview of selected methods enabling the analysis of the dynamics of large logical regulatory networks.

The definition of hierarchical transition graphs is at the core of Section III, referring to the relatively simple example of the bacteriophage lambda core regulatory network.

Section IV takes advantage of a recent comprehensive model of the regulatory network controlling T-helper cell differentiation in response to antigen presentation and to a dozen cytokines[27] to illustrate the power of the compression of state transition graphs into hierarchical transition graphs, as well as the insights gained into the corresponding logical dynamical behavior.

Finally, Section V proposes some global conclusions and discusses current challenges and further prospects.

II. LOGICAL MODELING AND ANALYSIS OF REGULATORY GRAPHS

This section introduces the basics of the logical formalism and presents a short overview of existing methods that enhance the dynamical analysis of logical models.

A. Logical regulatory graphs

A logical model is defined by an interaction graph where the nodes denote regulatory components (genes, proteins, etc.) and the arcs denote regulatory interactions. Moreover, a discrete variable is associated with each component, accounting for its level of activity (or expression). Logical functions define the dynamical evolution of the model.

Definition 1. A *Logical Regulatory Graph* $\mathcal{R} = (G, K)$ *is a graph, where*

- $G = \{g_i\}_{i=1,\ldots,n}$ *is the set of* n *regulatory components. Each component* g_i *is associated with a discrete variable* s_i *in* $D_i = \{0, \ldots, max_i\}$. *A state is thus defined as a vector* $s \in S = \Pi_{g_j \in G} D_j$.
- $K = \{K_i\}_{i=1,\ldots,n}$ *is the set of logical functions;* $K_i : S \to D_i$ *defines for each state, the target level of* g_i.

The arcs are deduced from the functions in K; there is a regulatory interaction from g_i *to* g_j *iff there are two states* s *and* s', *differing only by the value of* g_i, *that lead to different values of* K_i

$$\exists s, s' \in S \; s'_k = s_k \, \forall k \neq i, \text{ and } s_i \neq s'_i, \text{s.t.} \, K_j(s) \neq K_j(s').$$

Figure 1 illustrates this definition with a logical regulatory graph for the bacteriophage lambda switch.

The dynamics of logical models are represented in the form of state transition graphs as defined in Subsection III B.

B. State transition graphs

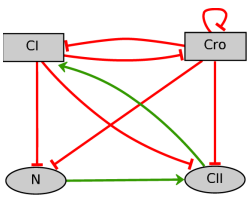
Definition 2. Given a logical regulatory graph $\mathcal{R} = (G, K)$, *its (full) STG, denoted by* $\mathcal{E} = (S, T)$, *is a directed graph with:*

- S *the state space of* $\mathcal{R} : S = \Pi_{g_j \in G} D_j$,
- $T : S^2 \to \{0, 1\}$ *the transition function: there is an arc connecting a state* s *to its successor* s' *whenever* $T(s, s') = 1$.

The transition function is defined according to an updating policy, which indicates the components to be updated in each transition. Here, for sake of brevity, we only consider the asynchronous updating policy (Definition 3). All results could be extended to other updating policies (including mixed (a) synchronous priority classes as presented in Sec. II C 2).

Definition 3. Given a logical regulatory graph $\mathcal{R} = (G, K)$, *the transition function defined according to the asynchronous updating policy specifies, for each state* s, *its successor states (as many as the components called to update in* s*):* $\forall (s, s') \in S^2$,

$$T(s, s') = \begin{cases} 1, \; if \; \exists g_i \in G \, s.t. \, K_i(s) \neq s_i, \\ \quad s'_i = s_i + \dfrac{|K_i(s) - s_i|}{K_i(s) - s_i} \; and \; \forall g_j \neq g_i, s'_j = s_j \\ 0, \; otherwise. \end{cases}$$



Logical functions	
$K_{CI}(s) = 2$	$\neg Cro \vee CII$
$K_{Cro}(s) = 3$	$\neg CI{:}2 \wedge \neg Cro{:}3$
$K_{Cro}(s) = 2$	$\neg CI{:}2 \wedge Cro{:}3$
$K_{CII}(s) = 1$	$\neg CI{:}2 \wedge \neg Cro{:}3 \wedge N$
$K_N(s) = 1$	$\neg CI \wedge \neg Cro{:}2$

FIG. 1. Logical regulatory graph of the bacteriophage lambda switch.[35] Left: the interaction graph, with the four components CI, Cro, CII, and N. Right: the logical functions, s denoting the vector $(s_{CI}, s_{Cro}, s_{CII}, s_N)$ of the component levels. For legibility, we rewrite each rule in terms of logical variables, e.g., CI denotes an interaction going out CI with a threshold 1, and it is true whenever $s_{CI} \geq 1$, while $CI:2$ denotes an interaction going out CI with a threshold 2; it is true whenever $s_{CI} \geq 2$. Here, for each component, we provide the rule(s) leading to a non-zero value of the logical function (meaning that when none of these conditions is fulfilled, the value is 0). For instance, the rule for $K_{CI}(s) = 2$ is satisfied for 30 states (those such that $s_{Cro} = 0$ or $s_{CII} = 1$); for all other states, CI's target value is 0. Note that values 1 of CI and Cro are always transient for this set of rules.

Note that updates are performed stepwise and thus transitions connect neighbouring states (i.e., their Hamming distance is 1).

We are often interested in a sub-graph of the full STG, which is generated considering a (set of) initial state(s). Then, the property of interest relates to the attractors reachable from this (set of) initial condition(s). Figure 2(a) displays the STG obtained for the phage lambda model, starting from a state where all variables are set to zero. Attractors, which denote asymptotical behaviors, are defined in a STG as its terminal *strongly connected components* (SCCs). Recall that strongly connected components are defined as the maximal strongly connected subgraphs (i.e., there is a path from each node to every other node).[5,13]

Given $\mathcal{E}=(S,T)$ a STG, we introduce further notation:

- Scc is the set of the SCCs of $\mathcal{E}$;
- $\forall s,s' \in S$, $s \rightsquigarrow s'$ means that there exists a path from s to s' (we consider that a sequence of a unique state forms a path of length 0, hence $s \rightsquigarrow s$);
- $\forall s \in S$, $\forall C \in Scc$, $s \rightsquigarrow C$ means that there exists a path from s to any state $s' \in C$;
- $\mathscr{S}$ is the set of the *trivial* SCCs (i.e., reduced to a single node) $\mathscr{S} = \{\{s\} \in Scc, s \in S\}$;
- $\mathscr{C}$ is the set of the *complex* SCCs: $\mathscr{C} = \{C \in Scc, \#C \geq 2\}$ (or $\mathscr{C} = Scc \backslash \mathscr{S}$);
- The sign * denotes *terminal* elements of Scc that will be referred to as *attractors*: C^* is terminal iff $\forall s \in C^*$, $\forall s' \in S$, $T(s,s') = 1 \Rightarrow s' \in C^*$. The non-terminal components are *transient*. In addition, $\mathscr{C}^*$ (resp. $\mathscr{S}^*$) denotes the set of the complex attractors (resp. the set of stable states).

Definition 4. Let A^* be an attractor, we define B_{A^*} the basin of attraction of A^* : $B_{A^*} = \{s \in S, s \rightsquigarrow A^*\}$. We further define $\bar{B}_{A^*}$, the strict basin of attraction of A^*,

$$\bar{B}_{A^*} = \{s \in B_{A^*}\ s.t. \forall X^* \in (\mathscr{C}^* \cup \mathscr{S}^*) \backslash \{A^*\},\ s \notin B_{X^*}\}.$$

Hence, A^* can be reached from any state in B_{A^*} or in $\bar{B}_{A^*}$; no other attractor can be reached from any state in $\bar{B}_{A^*}$.

C. Coping with large dynamics

Given a logical regulatory graph, the associated state space has $\prod_{g_i \in G} |D_i|$ elements (i.e., $2^{|G|}$, in the case of Boolean variables), meaning that its size grows exponentially with the number of regulatory components. Most properties are thus NP-complete, but one can mitigate this problem by lessening the size of the search space.

Here, we briefly review strategies to ease the analysis of large dynamics. A first approach consists in reducing the model, while ensuring the preservation of key properties. Another strategy lessens the number of transitions of the STG (hence simplifying the dynamics) assigning priorities to updating calls, relying on biologically well-founded assumptions. Other methods enable the reduction of the size of a STG, either by compacting it without losing any information, by applying appropriate reductions, or by considering alternate representations. Finally, we end this section with a short discussion on model-checking applied to multilevel logical models.

1. Model reduction

A first strategy to reduce the complexity of a model is to reduce its size, by removing some components. This is often done manually by the modeler, defining direct interactions even when it is known that the regulatory effects involve intermediate components. Obviously, by lessening the number of components, such reductions lead to smaller state spaces and hence simplified dynamics.

We have proposed to automate such model reductions and characterized their impact on the dynamics.[29] Basically, the reduction of a component amounts to attribute its regulatory role to its own regulators and to modify the logical function driving the behavior of its targets accordingly. The

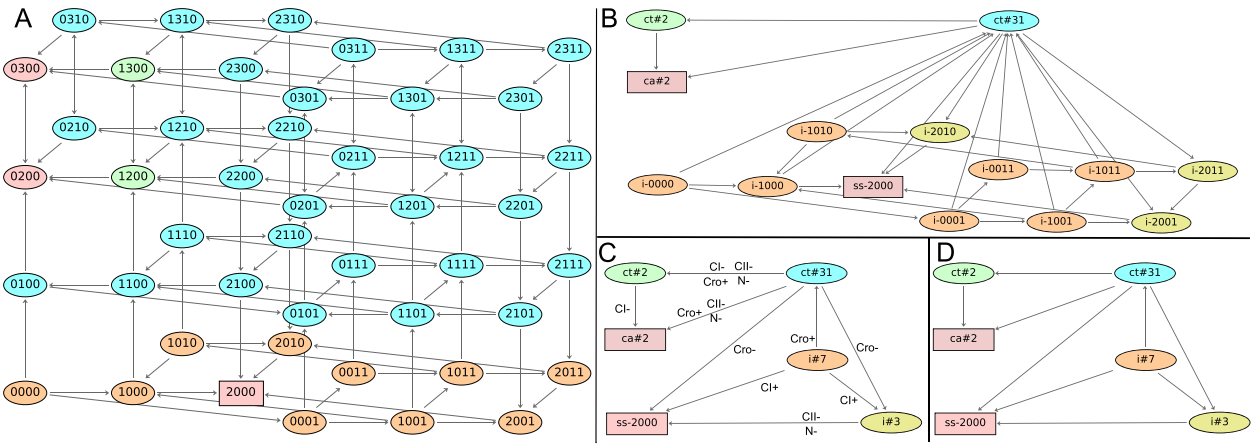


FIG. 2. Lambda phage model: different views of the dynamics. All the graphs, except that of panel (D), have been generated starting from the initial state (CI, Cro, CII, N) = 0000. (A) STG with the unique stable state indicated in a rectangular node; states sharing the same color belong to the same SCC. (B) Corresponding graph of the SCCs (same coloring as in panel A). (C) Corresponding HTG; arc labels refer to transitions in the underlying STG (i.e., updates of regulatory components). (D) HTG obtained using 1000 as initial state, which belongs to component i#7 in panel C; note the absence of a path from this state to the component i#3 (cf. remark 1). In the SCC and HTG graphs, node labels indicate the nature of the components: **i** irreversible; **ct** complex transient SCC; **ss** stable state; **ca** complex attractor; followed by the numbers of states in the component. For components reduced to a single state, the value of this state is displayed.

reduction of a self-regulated component is forbidden for it would not fit this rationale and would change the dynamical properties of the model. Indeed, we could prove that, provided this restriction on self-regulated components, the stable states and elementary terminal cycles of a reduced model exactly correspond to those of the original model. Moreover, the reduced model displays at least as many complex attractors, some corresponding to complex attractors of the original model, while others correspond to original transient oscillatory behaviors. In short, the main property of the proposed reduction is that it may suppress some transitions or paths, but never generates new ones.

Considering signaling networks including non-regulated input components, which usually account for external stimuli, Saadatpour *et al.* recently proposed to reduce input cascades that stabilize under constant input conditions.[32] This reduction has obviously no impact on the number and nature of the attractors, although it might change their reachability. Similarly, one can ignore (pseudo-)output components that have no outgoing interactions or that only regulate (pseudo-)output components.[28] Indeed, such output cascades have no impact, neither on the number and nature of the attractors, nor on their reachability.

2. Priority classes

Asynchronous state transitions graphs can be sometimes simplified by reducing the number of transitions, using relatively simple temporal assumptions. Indeed, in all states, the asynchronous scheme defines as many transitions as the number of components called to update, thus potentially generating spurious trajectories. A number of these can be ignored by defining priority classes ranking updating calls.[14] When two calls with distinct priority ranks are enabled in a state, the one with the lowest rank is discarded. Updatings belonging to the same class can be treated synchronously or asynchronously. In GINsim, it is thus possible to partition component updatings into distinct classes that implement such a priority scheme.[9,26] Needless to say, priority classes should be biologically well-founded to ensure that discarded trajectories are indeed irrelevant.

3. Lessening the size of the state transition graph

Several studies have addressed the problem of the combinatorial explosion of the state space of asynchronous transition systems.

Given a STG, an informative view of the dynamics is provided by the graph of its SCCs, where each node accounts for one SCC (possibly keeping the information of the states it encompasses). The resulting graph is a directed acyclic graph, which is often much smaller than the original STG, yet keeping all the reachability information (see Figures 2(a) and 2(b)). Tarjan defined an efficient algorithm to compute the SCCs of a directed graph (linear in the number of nodes and arcs).[34] Tournier and Chaves[39] have already applied SCC decomposition to STGs. However, SCC compaction remains limited in the case of networks with long or numerous regulatory cascades, which give rise to multiple linear (although potentially branching) pathways in the resulting STGs.

Another approach that keeps the whole STG structure applies to models that encompass a significant number of *input* components. These account for external stimuli (e.g., environmental cues) and the corresponding variables are generally maintained constant. In this case, the STGs corresponding to different combinations of input values are disconnected. Input components may also be considered as "uncontrolled" variables, which are allowed to freely vary at each time step. A natural reduction consists in projecting the state space on the set of internal components and labelling each transition with the values of the input components that enable that transition (for more details, see Ref. 24).

Other strategies, mainly developed by the formal verification community, reduce the state space yet ensuring that truth values of (linear) temporal logic formulas are preserved. This is the case of partial-order reduction methods that basically consist in identifying, for each state, a subset of transitions to explore (hence not exploring all the successors). Alternative (rather similar) definitions of these sets have been proposed, called stubborn, ample, or persistent sets.[12,19]

Relying on the Petri net representation of logical regulatory graphs[8] and using Petri net tools (e.g., TINA[6]), we have recently applied such a partial-order reduction to check reachability properties on a large logical model (encompassing 72 regulatory components). For this specific model, due to the structure of its dynamics, partial-order reduction proved to be poorly effective. However, there is certainly room for improvements of these methods,[16] and further work might identify a class of logical graphs more amenable to this kind of reductions.

4. Model-checking

During the recent years, formal verification techniques based on model-checking have been successfully applied to the analysis of molecular network models.[4,7,24,25] This approach is directly applicable to the verification of logical regulatory graphs, which constitute a class of finite state systems. In general, experimentally observed biological behaviors can be expressed in terms of temporal logic statements, and model-checking algorithms used to automatically verify if a model satisfies these statements.

When using explicit representations of states and transitions, model-checking may use partial-order reduction to lessen the size of the search space. However, symbolic model-checking relies on implicit representations, scaling better for large models. The choice of the temporal logic depends on the type of property to be checked.[12] Here, we are mainly interested in attractor reachability from a (set of) initial condition(s) as well as in the conditions enabling such trajectories. This supposes a previous characterization of the attractors, among which the stable states can be efficiently identified beforehand.[9]

GINsim includes an export converting logical models into NuSMV symbolic descriptions.[24] NuSMV is a symbolic model-checking tool capable of verifying finite state

machines against a set of property requirements, expressed as temporal logic formulas.[11] This export supports the definition of priority classes and takes advantage of the reduction over input components evoked in Sec. II C 3, these being specified either as constant or as freely varying variables. Noteworthy, in the case of varying inputs, the notion of stable states needs to be extended: a state may be stable for given values of input components, and not for others.[28] For models with input components, it is thus possible to analyze switches between cellular types (i.e., stable states) and verify the corresponding input component variations (see Sec. IV and Figure 6).

III. HIERARCHICAL TRANSITION GRAPH

This section deals with the definition and properties of a novel, compact hierarchical graph, where a set of states is shrunken into a single node, whenever it forms a Strongly Connected Component (SCC), or a (set of) linear chain(s) leading to the same set of SCCs and attractors. Compared to the SCC graph mentioned above, this graph generally corresponds to a further reduction of the State Transition Graph (STG). Furthermore, the resulting grouping of states greatly eases the interpretation of the structure of the dynamics in terms of basins of attraction.

A. Definitions

Let us first define the application σ that associates to each SCC C, the set of SCCs, complex or terminal, that are reachable from C, including C itself if it is complex or terminal

$$\sigma(X) = \{C \in \mathscr{C} \cup \mathscr{S}^* \text{ s.t. } X = C \text{ or } \forall s \in X,\ s \rightsquigarrow C\}.$$

Furthermore, we define $\mathcal{I} \subset 2^S$, the set of *irreversible transient components* in which trivial non-terminal SCCs (elements of $\mathscr{S} \backslash \mathscr{S}^*$) that have the same σ-image are grouped together

$$\mathcal{I} = \{I \in 2^S \text{ s.t. } \forall s \in I,\ \{s\} \in \mathscr{S} \backslash \mathscr{S}^* \text{ and } s, s' \in I \Rightarrow \sigma(\{s\}) = \sigma(\{s'\})\}.$$

Definition 5. Given a STG $\mathcal{E} = (S, T)$, *we denote* $\mathcal{H} = (\mathcal{C} \cup \mathcal{I} \cup \mathcal{S}^*, \mathcal{T})$ *its corresponding Hierarchical Transition Graph (HTG), where* $\mathcal{T} : (\mathcal{C} \cup \mathcal{I} \cup \mathcal{S}^*)^2 \to \{0, 1\}$ *defines the arcs of* $\mathcal{H}$

$$\mathcal{T}(C, C') = 1 \iff \exists s \in C,\ \exists s' \in C' \ s.t.\ T(s, s') = 1.$$

Each complex SCC of the STG is contracted to a single node in the HTG. Similarly, a single HTG node accounts for all trivial SCCs sharing the same σ-image. Figure 2 provides an illustration of the HTG construction for the lambda phage model.

B. Properties

For two components $C, C' \in \mathscr{C} \cup \mathcal{I} \cup \mathscr{S}^*$, the notation $C \overset{\mathcal{H}}{\rightsquigarrow} C'$ indicates the existence of a path from C to C' in the HTG. The following property relates paths in the STG to paths in the corresponding HTG.

Property 1.

1. *A path connecting any HTG component to a non-irreversible component implies the existence of a path in the corresponding STG*

$$\forall C \in \mathscr{C} \cup \mathcal{I},\ C' \in \mathscr{C},\quad C \overset{\mathcal{H}}{\rightsquigarrow} C' \Rightarrow s \rightsquigarrow s' \quad \forall s \in C, \forall s' \in C'.$$

2. *A path between two states in the STG implies the existence of a path between the HTG components they belong to*

$$\forall s, s' \in S, s \rightsquigarrow s' \Rightarrow C \overset{\mathcal{H}}{\rightsquigarrow} C', \quad \text{with} \quad s \in C, s' \in C'.$$

Proof.

1. Let $C \overset{\mathcal{H}}{\rightsquigarrow} C'$, with $C \in \mathscr{C} \cup \mathcal{I}$ and $C' \in \mathscr{C} \cup \mathscr{S}^*$. Then, $C' \in \sigma(C) : \forall s \in C,\ s \rightsquigarrow C'$ and the first item of Property 1 is proved by definition.
2. Let $s, s' \in S$ s.t. $s \rightsquigarrow s'$, and denote C and C' the components of the HTG, such that $s \in C$ and $s' \in C'$.
   - If $C = C'$, the statement is obviously true.
   - If $C \neq C'$, let $(s = s_1, s_2, \ldots, s_k = s')$ be the path from s to s' in the STG: $\forall i = 1, \ldots, k-1,\ s_i \in S$ and $T(s_i, s_{i+1}) = 1$; if $\forall i = 1, \ldots, k$, C_i denotes the component of Scc such that $s_i \in C_i$, we have $\mathcal{T}(C_i, C_{i+1}) = 1$ or $C_i = C_{i+1}$. Hence, following the path $s \rightsquigarrow s'$, we obtain that $C \overset{\mathcal{H}}{\rightsquigarrow} C'$.

Remark 1. Property *1 does not ensure equivalence of path existence in STG and related HTG. Indeed, in item 1, we have the restriction that* $C' \notin \mathcal{I}$: *when* $C \overset{\mathcal{H}}{\rightsquigarrow} C'$, *with* $C' \in \mathcal{I}$, *given* $s \in C$, *we cannot ensure the existence of a path in the STG from s to a state* $s' \in C'$.

In Figure 2, we have such a situation, where $C \overset{\mathcal{H}}{\rightsquigarrow} C'$, *with* $C' \in \mathcal{I}$ *and* $\exists s \in C$ *s.t., there is no path in the STG from s to any state in* C'. *Indeed, considering Figure 2, panel C, the irreversible component i#7 contains the state 1000 (see panel B), and the arc from i#7 to i#3 indicates that there exists a state* $s \in$ *i#7 and a state* $s' \in$ *i#3 such that* $s \rightsquigarrow s'$ *(e.g., T(1011, 2011) = 1, in panel A or B). However, there is no path from state 1000 to any state of i#3 as illustrated in panel D.*

Another typical situation for which we have $C \overset{\mathcal{H}}{\rightsquigarrow} C'$, $C' \in \mathcal{I}$, *and no path in the STG from* $s \in C$ *to* $s' \in C'$, *may arise when a hierarchical (irreversible) component contains disconnected states.*

We propose an algorithm to generate HTGs of logical regulatory graphs, given a (set of) initial condition(s). Described in the supplementary file,[43] this algorithm is based on Tarjan's method[34] and compacts a STG on-the-fly.

C. Basins of attraction

A classical way to study the dynamics is to focus on attractors and their basins of attraction (cf. Definition 4). When using the synchronous dynamics, their computation is facilitated by the fact that all states have at most one successor (for more details, see Ref. 42). But in the case of concurrent behavior, it is computationally much more costly (see Refs. 1, 39, and 41 for the fully asynchronous case).

By construction, the HTG nodes group together states of the STG and thus allow to easily recover the basins of attractions. Indeed, given A^* an attractor in the STG, and $C \in \mathscr{C} \cup \mathscr{I} \cup \mathscr{S}^*$ a node of the HTG, the states of C are in B_{A^*}, the *basin of attraction* of A^*, iff $A^* \in \sigma(C)$. The states of C are in $\bar{B}_{A^*}$, the *strict basin of attraction* of A^*, iff $\sigma(C) \cap (\mathscr{C}^* \cup \mathscr{S}^*) = \{A^*\}$. Hence, for all attractors, it is much easier to identify their basins of attraction on the basis of the HTG.

Irreversible decisions are taken at the intersections of the basins of attraction. Given two consecutive nodes X_1 and X_2 in the HTG ($\mathcal{T}(X_1, X_2) = 1$), crucial decisions can be associated with the arc linking these nodes if $\sigma(X_1) \cap (\mathscr{C}^* \cup \mathscr{S}^*) \neq \sigma(X_2) \cap (\mathscr{C}^* \cup \mathscr{S}^*)$ (i.e., the system enters a more restricted basin of attraction). Consider two attractors A_1^* and A_2^* such that $A_1^* \in \sigma(X_1) \cap \sigma(X_2)$, $A_2^* \in \sigma(X_1)$ and $A_2^*/\sigma(X_2)$. We say that the transition (X_1, X_2) belongs to the *boundary* of $B_{A_2^*}$: removing all such transitions would isolate $B_{A_2^*}$ from the rest of the HTG. In Figure 2(c), transitions (ct#31, ct#2) and (ct#31, ca#2) constitute the boundary of $B_{\mathrm{ss}-2000}$.

IV. APPLICATION TO Th CELL DIFFERENTIATION

T-helper lymphocytes play a key role in the regulation of the immune response in vertebrate. Various T-helper subtypes (Th1, Th2, Th17, Treg) have been identified over the years, characterized by the expression of specific transcription factors and cytokines, which have a critical influence on the selection of specific immune responses, driving pro-inflammatory or allergic responses, promoting alternative antibody classes, or yet preventing (auto)immunity by inhibiting the activation and proliferation of other cells.

Several modeling studies have been proposed to shed light on the regulatory network controlling T-helper cell activation and differentiation (see, e.g., Refs. 20, 21, 23, 33, 40 and references therein).

To gain insight into the heterogeneity and the plasticity of late T-helper lineages, we have recently built an integrated logical model of the core regulatory network and main signaling pathways controlling Th cell differentiation[27] (Figure 3). Encompassing 65 components (including 13 inputs, corresponding to antigen presentation and a dozen different cytokines), this multilevel logical model proved to be too complex to be straightforwardly simulated. This situation motivated the development of the reduction method mentioned above.[27,29] In the case of our T-helper model, we were led to hide 31 components (shown in grey in Figure 3) and thus obtained a more compact model encompassing 34 nodes (including the same 13 inputs).

Using the resulting reduced T-helper model, we have performed a series of simulations to assess the effects of heterogeneous environments on Th cell differentiation. This led us to identify stable states corresponding to canonical Th1, Th2, Th17, and Treg subtypes, but also to hybrid cell types co-expressing combinations of Th1, Th2, Treg, and Th17 markers in an environment-dependent fashion.

Here, we apply the HTG construction to this reduced model in order to demonstrate how the dynamics can be compressed in a meaningful way, emphasizing the structure of the underlying STG, as well as crucial decision points along dynamical pathways. In this respect, we have selected a limited number of simulations leading to STGs of increasing complexity.

Figure 4 displays the HTGs obtained when simulating a naive T-helper cell stimulated by an antigen presenting cells in the presence of IL2 alone, or in the presence of pro-Th1, Th2, Treg, or Th17 cytokines. In all cases but the last one, we obtain a unique stable state corresponding to the expected cellular state (activated Th0, Th1, Th2, or Treg). In each of these HTGs, all other states reachable from the initial conditions are grouped together into a single irreversible transient component, encompassing between 25 and 255 states. The label associated with each arc denotes the ultimate elementary transitions going out the HTG node. In contrast, in pro-Th17 conditions, the system can reach two different stable states expressing Th17 transcription factor RORGT, IL10, IL21, and IL23, one expressing also FOXP3, the other expressing IL2. From the arc labels, it follows that the selection between these two stable states depends on the concurrent activation of RORGT and FOXP3.

Figure 5 (top) displays the HTG obtained when simulating a naive T-helper cell stimulated in mixed pro-Th2/pro-

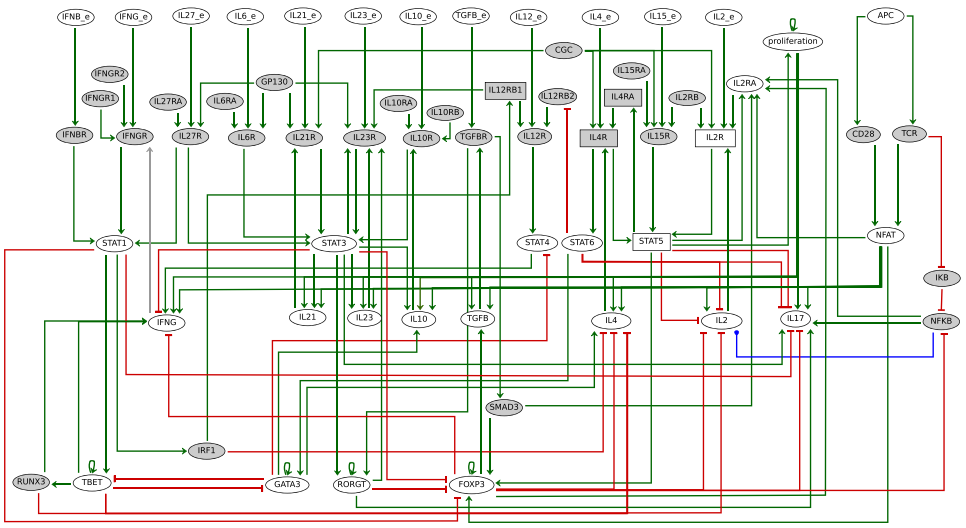


FIG. 3. Th differentiation regulatory graph. The top nodes correspond to inputs (APC and external input cytokines), while nodes placed at the bottom correspond to key transcription factors. Nodes considered for reduction are emphasized in grey. Green arrows denote activations, red blunt arcs denote inhibitions while the blue arc from NFKB to IL17 denotes a dual interaction (see Ref. 27 for details).

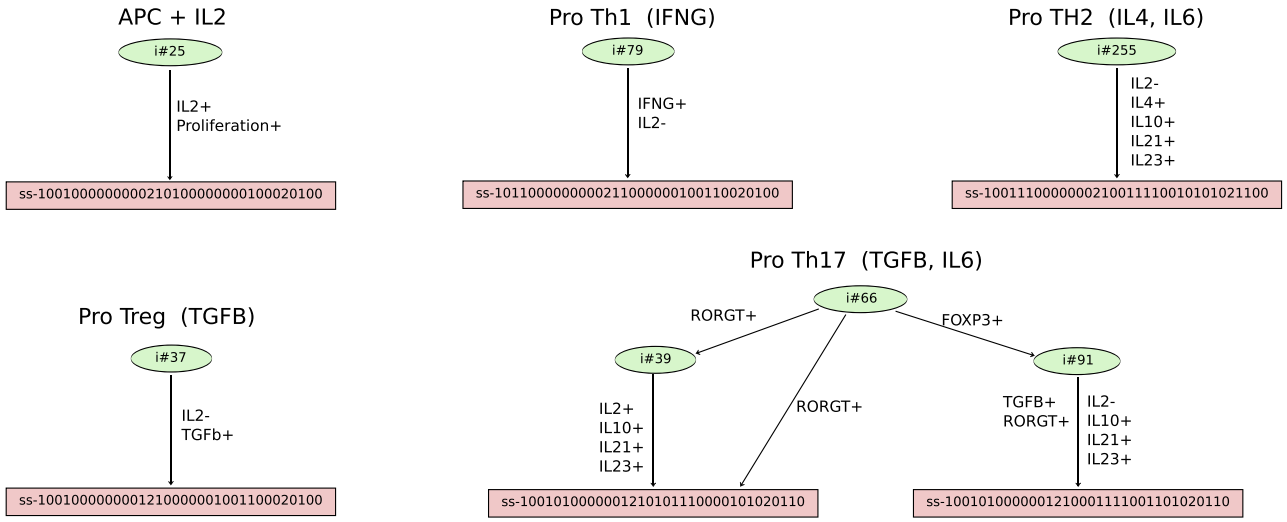


FIG. 4. HTG representation of asynchronous simulations of naive Th cells in simple polarizing environments. These HTGs correspond to the simulation of Th0 cells in the presence of APC signalling + IL2 alone, with INFG (pro Th1), or with IL4 + IL6 (pro Th2), or with TGFB (pro Treg), or with TGFB and IL6 (pro Th17), from top to bottom and left to right. All other nodes are set to zero at the initial state. In the notation of the logical stable states (prefixed by "ss-"), the node order considered starts with APC, followed by the external input cytokines IFNB, INFG, IL2, IL4, IL6, IL10, IL12, IL15, IL21, IL23, IL27, TGFB, followed by the receptor components IL2R and IL2RA, and then by the cytokines produced by the Th cell considered INFG, IL2, IL4, IL10, IL21, IL23, TGFB, then the transcription and signal transduction factors TBET, GATA3, FOXP3, NFAT, STAT1, STAT3, STAT4, STAT5, STAT6, followed by the proliferation node, followed by RORGT and IL17. Arc labels indicate transitions (regulatory component updates) driving the system out of an HTG node toward another one.

Th17 environment, i.e., in the presence of IL4, IL6, TGFB, and in the absence of IL2. The resulting HTG merely comprises 13 nodes, to be contrasted with the 1146 states of the corresponding STG. Furthermore, the HTG structure emphasizes the progressive commitment of cells when following paths from the root to the leaves (stable states). The states encompassed by other nodes belong to two or more basins of attraction. Note that the system can reach four stable states, more precisely two pairs of activated versus anergic Th2 RORGT+ subtypes. Within each of these pairs, the stable states differ by the expression level of FOXP3. The labels associated with the arcs clearly emphasize the transitions implementing differentiation decisions. As illustrated in Figure 5 (bottom), the use of priorities significantly decreases the size of the dynamics; selecting updates of ILR2, NFAT, and any of the STAT factors against other component updates led to an HTG of 5 nodes (encompassing 31 states) instead of 13 nodes (encompassing 1146 states), where the two anergic cellular types are the only reachable stable states.

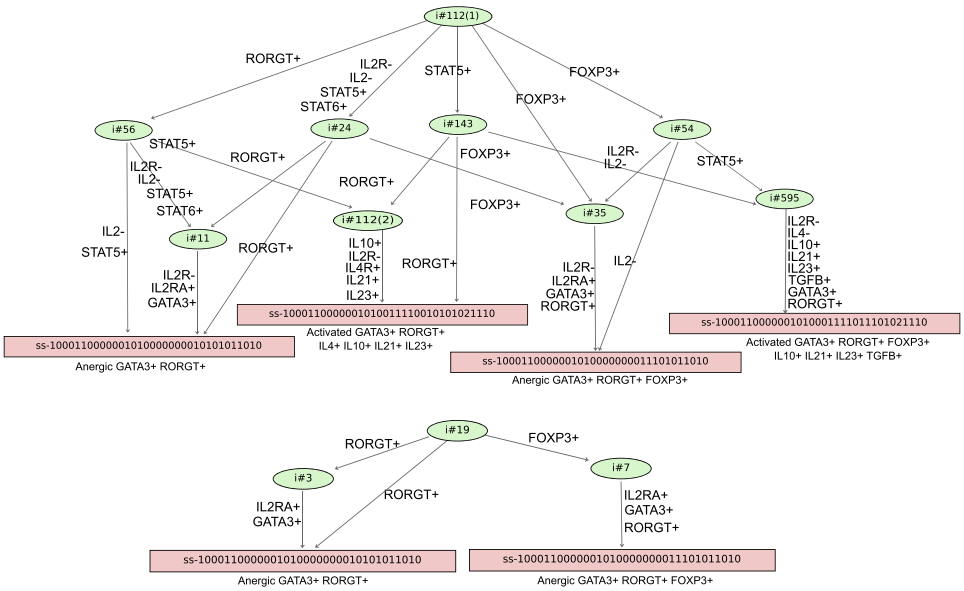


FIG. 5. Compressed representation (HTG) of the asynchronous simulation of Th0 in the presence of APC signaling + IL4 + IL6 + TGFB (combination of pro Th2 and Th17 cytokines, in the absence of IL2). Four stable states can be reached: two pairs of activated versus anergic Th2 RORGT+ cells, differing by the expression of FOXP3. The bottom part shows the HTG obtained for the same initial conditions but using two asynchronous priority classes. In this configuration, transitions involving IL2R, NFAT, or any of the STAT factor are selected against those involving any other component. In contrast with the results obtained without prioritization, only two stable states can be reached, both corresponding to anergic Th2 RORGT+ cells, which differ by the expression of FOXP3. The labels associated with the arcs emphasize the crucial transitions underlying the choice of one or the other differentiation state.

A thorough discussion of the biological significance of these observations would go beyond the scope of this article. However, these examples demonstrate the compression and clarification of asynchronous simulations that can be achieved using the HTG representation.

Finally, Figure 6 displays the reachability analysis between cellular types through the use of model-checking. For this, we considered the environmental conditions defined by specific valuations of the 13 input components (see Figure 5 of the original study, i.e., in Ref. 27) and the stable expression patterns (see Figure 6 in Ref. 27). These stable states correspond to cellular subtypes, which are stable under specific environmental conditions. So, for each input combination, we verify the existence of a direct path between each possible pair of cellular types. More precisely, we check whether there is a fixed valuation of inputs such that there is a direct path between two cellular subtypes, $C1$ and $C2$ (without going through other cellular subtypes) and $C2$ being stable. Using this approach, we could reproduce the results obtained at the cost of extensive simulations in the original study (Figure 7 in Ref. 27). Three main groups are defined over the Th cell subtypes (Th0, Th1, and Th2, see Figure 8 in Ref. 27). We could also verify that Th1 and Th2 subtypes can never switch back to a Th0 one, even when inputs are allowed to vary freely. However, in such case, switches between all cellular subtypes are possible within each group.

V. CONCLUSION AND PROSPECTS

HTGs emphasize relevant transient and asymptotic dynamical properties. We have defined a novel algorithm enabling the compaction of state transition graphs and the generation of HTGs on-the-fly. This approach has been implemented into a development version of the software GINsim, available as a pre-release.[30]

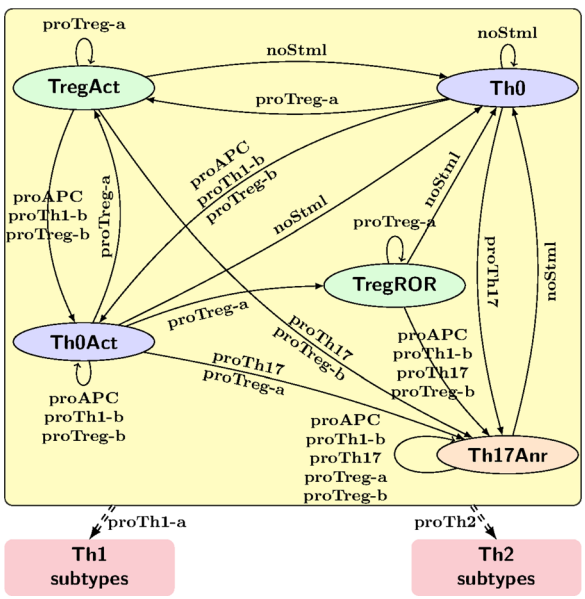


FIG. 6. Model-checking reachability analysis between cellular types under fixed input conditions using NuSMV.[11] Nodes represent cellular subtypes (see Figure 6 in Ref. 27), whereas arrows represent the existence of a direct path between two cellular types under a specific fixed environmental condition (see Figure 5 in Ref. 27). For simplicity, reachability analysis has been truncated to the Th0 subtypes, discarding the Th1 and Th2 subtypes.

We have applied this approach to a comprehensive model for T-helper cell differentiation. Although this model still needs to be further refined and tested, the analysis presented here clearly demonstrates the assets of the HTG representation, which leads to significant graph compression and clearly emphasizes the organization of the state space into attractors and basins of attraction.

Interestingly, applying our algorithm for HTG construction onto a HTG produces a further compacted graph-based representation of the dynamics, where the nodes correspond to basins of attraction.

Should a given dynamics be too large and complex to be effectively compacted using the HTG representation, we can rely on complementary methods presented in this manuscript. These methods aim at reducing the size of the search space, including the model reduction method that preserves key dynamical properties and the definition of transition priority classes, relying on biologically well-founded assumptions. Moreover, we have presented model-checking techniques to analyse reachability properties. Used jointly, these methods enable the dynamical analysis of logical models of unprecedented sizes.

It is worth noting that the HTG structure could be considered in the context of other formal approaches relying on state transition graphs, including Petri nets (see, e.g., Ref. 8 and references therein) and piecewise-linear differential equation (PLDE) models.[3,17] Model-checking techniques also apply to these models, once their dynamics can be represented by Kripke structures.[12] Our model reduction could be applied to PLDE models, but its impact on the dynamics still needs to be clarified.

HTG construction could be optimized and improved, e.g., using parallel algorithms. Although depth-first search algorithms are known to be difficult to parallelize,[31] different methods have been proposed to tackle this problem.[2]

Further analysis relying on HTG structures should allow the assessment of finer properties. For instance, some well-established rules (Thomas' rules[36]) assert that differentiation (resp. homeostasis) phenomena lean on the action of a positive (resp. negative) circuit in the regulatory graphs. In practice, circuit functionality analysis often points to combinations of intertwined circuits, which are difficult to analyze. HTGs appear particularly well suited to the dynamical analysis of complex networks endowed with differentiation properties (i.e., presenting multiple alternative stable states, which can be all reached from given initial conditions), as they capture the general organization of the corresponding STGs. Hence, based on the analysis of HTG structures, we should be able to identify the circuits at the core of cell commitment and thereby focus on the genes responsible for irreversible decisions.

An alternative strategy to analyze large regulatory networks takes advantage of their modularity. Recently, we have defined a compositional framework that relies on process algebra to incrementally compose, abstract, and minimize (using the safety equivalence) logical regulatory modules, enabling impressive reductions of the dynamics.[22] However, as proper methods to decompose large networks

into functional modules are still lacking, we have focussed on regulatory networks that encompass several identical modules connected by inter-cellular signaling. In this respect, one could take advantage of HTG structures to come up with relevant model decomposition.

ACKNOWLEDGMENTS

We acknowledge support from the EU FP7 (APOSYS large scale project), the EU EraSysBio+ program (project ModHeart) the ANR (Project Grant ANR-08-SYSC-003), FCT (Project Grant PTDC/EIA-CCO/099229/2008), and the Belgian Science Policy Office (IAP BioMaGNet).

[1]J. Bahi and S. Contassot-Vivier, "Basins of attraction in fully asynchronous discrete-time discrete-state dynamics networks," AIEEE Trans. Neural Networks **17**, 397–408 (2006).
[2]J. Barnat, J. Chaloupka, and J. van de Pol, "Distributed algorithms for SCC decomposition," J. Logic Comput. **21**, 23–44 (2011).
[3]G. Batt, B. Besson, P.-E. Ciron, H. de Jong, E. Dumas, J. Geiselmann, R. Monte, P. T. Monteiro, M. Page, F. Rechenmann, and D. Ropers, "Genetic network analyzer: A tool for the qualitative modeling and simulation of bacterial regulatory networks," Methods Mol. Biol. **804**, 439–462 (2012).
[4]G. Batt, D. Ropers, H. de Jong, J. Geiselmann, R. Mateescu, M. Page, and D. Schneider, "Validation of qualitative models of genetic regulatory networks by model checking: Analysis of the nutritional stress response in *Escherichia coli*," Bioinformatics **21**, i19–i28 (2005).
[5]C. Berge, *Graphs and Hypergraphs*, North-Holland Mathematical Library, Vol. 6 (North-Holland Pub. Co., Amsterdam, 1973).
[6]B. Berthomieu, P.-O. Ribet, and F. Vernadat. "The tool TINA—construction of abstract state spaces for Petri nets and time Petri nets," Int. J. Prod. Res. **42**, 2741–2756 (2004).
[7]N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter, "Modeling and querying biomolecular interaction networks," Theor. Comput. Sci. **325**(1), 25–44 (2004).
[8]C. Chaouiya, A. Naldi, E. Remy, and D. Thieffry, "Petri net representation of multi-valued logical regulatory graphs," Nat. Comput. **10**, 727–750 (2011).
[9]C. Chaouiya, A. Naldi, and D. Thieffry, "Logical modelling of gene regulatory networks with GINsim," Methods Mol. Biol. **804**, 463–479 (2012).
[10]C. Chaouiya, E. Remy, B. Mossé, and D. Thieffry, "Qualitative analysis of regulatory graphs: A computational tool based on a discrete formal framework," Lect. Notes Control Inf. Sci. **294**, 119–126 (2003).
[11]A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella, "NuSMV2: An OpenSource tool for symbolic model checking," Lect. Notes Comput. Sci. **2404**, 359–364 (2002).
[12]E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking* (MIT Press, 1999).
[13]T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms* (MIT Press, 1990).
[14]A. Fauré, A. Naldi, C. Chaouiya, and D. Thieffry, "Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle," Bioinformatics **22**, e124–e131 (2006).
[15]A. Fauré, A. Naldi, F. Lopez, C. Chaouiya, A. Ciliberto, and D. Thieffry, "Modular logical modelling of the budding yeast cell cycle," J. Mol. BioSyst. **5**, 1787–1796 (2009).
[16]J. Geldenhuys, H. Hansen, and A. Valmari, "Exploring the scope for partial order reduction," Lect. Notes Comput. Sci. **5799**, 39–53 (2009).
[17]L. Glass and S. A. Kauffman, "The logical analysis of continuous, non-linear biochemical control networks," J. Theor. Biol. **39**, 103–129 (1973).
[18]L. Glass and H. T. Siegelmann, "Logical and symbolic analysis of robust biological dynamics," Curr. Opin. Genet. Dev. **20**(6), 644–649 (2010).
[19]P. Godefroid (ed.), "Partial-order methods for the verification of concurrent systems—an approach to the state-explosion problem," Lect. Notes Comput. Sci. **1032** (1996).
[20]T. Hong, J. Xing, L. Li, and J. Tyson, "A simple theoretical framework for understanding heterogeneous differentiation of CD4+T cells," BMC Syst. Biol. **6**, 66 (2012).
[21]S. Klamt, J. Saez-Rodriguez, J. A. Lindquist, L. Simeoni, and E. D. Gilles, "A methodology for the structural and functional analysis of signaling and regulatory networks," BMC Bioinform. **7**, 56 (2006).
[22]N. D. Mendes, F. Lang, Y.-S. Le Cornec, R. Mateescu, G. Batt, and C. Chaouiya, "Composition and abstraction of logical regulatory modules: Application to multicellular systems," Bioinformatics **29**, 749–757 (2013).
[23]L. Mendoza and F. Pardo, "A robust model to describe the differentiation of T-helper cells," Theory Biosci. **129**, 283–293 (2010).
[24]P. T. Monteiro and C. Chaouiya, "Efficient verification for logical models of regulatory networks," Adv. Intell. Soft Comput. **154**, 259–267 (2012).
[25]P. T. Monteiro, D. Ropers, R. Mateescu, A. T. Freitas, and H. de Jong, "Temporal logic patterns for querying dynamic models of cellular interaction networks," Bioinformatics **24**, i227–i233 (2008).
[26]A. Naldi, D. Berenguier, A. Fauré, F. Lopez, D. Thieffry, and C. Chaouiya, "Logical modelling of regulatory networks with GINsim 2.3," Biosystems **97**, 134–139 (2009).
[27]A. Naldi, J. Carneiro, C. Chaouiya, and D. Thieffry, "Diversity and plasticity of Th cell types predicted from regulatory network modelling," PLoS Comput. Biol. **6**, e1000912 (2010).
[28]A. Naldi, P. T. Monteiro, and C. Chaouiya, "Efficient handling of large signalling-regulatory networks by focusing on their core control," Lect. Notes Comput. Sci. **7605**, 288–306 (2012).
[29]A. Naldi, E. Remy, D. Thieffry, and C. Chaouiya, "Dynamically consistent reduction of logical regulatory graphs," Theor. Comput. Sci. **412**, 2207–2218 (2011).
[30]See http://ginsim.org/beta for a beta version of GINsim that includes the construction of Hierarchical Transition Graphs.
[31]J. H. Reif, "Depth-first search is inherently sequential," Inform. Process. Lett. **20**, 229–234 (1985).
[32]A. Saadatpour, I. Albert, and R. Albert, "Attractor analysis of asynchronous Boolean models of signal transduction networks," J. Theor. Biol. **266**, 641–656 (2010).
[33]J. Saez-Rodriguez, L. Simeoni, J. A. Lindquist, R. Hemenway, U. Bommhardt, B. Arndt, U.-U. Haus, R. Weismantel, E. D. Gilles, S. Klamt, and B. Schraven, "A logical model provides insights into T cell receptor signaling," PLoS Comput. Biol. **3**, e163 (2007).
[34]R. Tarjan, "Depth-first-search and linear graph algorithms," SIAM J. Comput. **1**, 146–160 (1972).
[35]D. Thieffry and R. Thomas, "Dynamical behaviour of biological regulatory networks–II. Immunity control in bacteriophage lambda," Bull. Math. Biol. **57**, 277–297 (1995).
[36]R. Thomas, "On the relation between the logical structure of systems and their ability to generate multiple steady states and sustained oscillations," Ser. Synergetics **9**, 180–193 (1981).
[37]R. Thomas and R. D'Ari, *Biological Feedback* (CRC Press, 1990).
[38]R. Thomas, D. Thieffry, and M. Kaufman, "Dynamical behaviour of biological regulatory networks–I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state," Bull. Math. Biol. **57**, 247–276 (1995).
[39]L. Tournier and M. Chaves, "Uncovering operational interactions in genetic networks using asynchronous Boolean dynamics," J. Theor. Biol. **260**, 196–209 (2009).
[40]H. J. van den Ham and R. J. de Boer, "From the two–dimensional Th1 and Th2 phenotypes to high-dimensional models for gene regulation," Int. Immunol. **20**, 1269–1277 (2008).
[41]K. Willadsen and J. Wiles, "Robustness and state-space structure of Boolean gene regulatory models," J. Theor. Biol. **249**, 749–765 (2007).
[42]A. Wuensche, "Complex and chaotic dynamics, basins of attraction, and memory in discrete networks," Acta Phys. Pol. B - Proc. Suppl. **3**, 463–478 (2010), available at: http://th-www.if.uj.edu.pl/acta/sup3/abs/s3p0463.htm.
[43]See supplementary material at http://dx.doi.org/10.1063/1.4809783 for details on the algorithm that constructs a hierarchical transition graph.