Theoretical Computer Science 412 (2011) 2207–2218

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

# Dynamically consistent reduction of logical regulatory graphs

Aurélien Naldi [a,*], Elisabeth Remy [b], Denis Thieffry [a,c], Claudine Chaouiya [a,d]

[a] TAGC Inserm U928 - Université de la Méditerranée, Marseille, France
[b] IML UMR 6206, Marseille, France
[c] CONTRAINTES project, INRIA Paris-Rocquencourt, France
[d] IGC, Instituto Gulbenkian de Ciência, Oeiras, Portugal

ARTICLE INFO

*Keywords:*
Regulatory networks
Logical modelling
Model reduction
Decision diagrams
Attractors
Drosophila segmentation

ABSTRACT

To cope with the increasing complexity of regulatory networks, we define a reduction method for multi-valued logical models.

Starting with a detailed model, we use decision diagrams to compute reduced models by iteratively "removing" regulatory components. To keep a consistent dynamical behaviour, the logical rules associated with the targets of each removed node are actualised to account for the (indirect) effects of its regulators.

This construction of reduced models preserves crucial dynamical properties of the original model, including stable states and more complex attractors. In this respect, the relationship between the attractor configuration of the original model and those of reduced models is formally established. We further analyse the issue of attractor reachability.

Finally, we illustrate the flexibility and efficiency of the proposed reduction method by its application to a multi-valued model of the fly segment polarity network, which is involved in the control of segmentation during early embryogenesis.



## 1. Introduction

Massive data generation and integration efforts result in the delineation of ever more comprehensive and complex regulatory networks involved in the control of numerous biological processes. Consequently, current modelling and analysis approaches are reaching their limits in terms of the number and variety of components and interactions that can be efficiently considered. This is true for quantitative frameworks (*e.g.*, differential or stochastic models), as well as for qualitative approaches. Indeed, although logical modelling enables the handling of networks comprised of relatively large numbers of components (see *e.g.* [1,2]), the size of the state space grows exponentially with the number of regulatory nodes.

One way to handle this problem consists in developing compositional approaches to compute the dynamical properties of comprehensive networks, relying on the knowledge of the properties of simpler sub-systems or modules. A complementary approach consists in reducing large systems, by focusing on the most relevant components and redefining their interactions in order to preserve relevant dynamical properties (*e.g.* stable states).

Most often, modellers intuitively and manually reduce regulatory networks to address specific questions. Such empirical reductions have several drawbacks: (i) the process is error prone and limited to relatively simple cases; (ii) the maintenance of different versions of a model (complete and reduced) is cumbersome; (iii) storing the sole reduced model leads to the loss of relevant biological information.

These considerations led us to develop a reliable, automated reduction method in the context of a logical modelling framework. Established on firm mathematical bases and implemented into the logical modelling software *GINsim* [3,4], our

* Corresponding author. Tel.: +33 491 82 87 22.
*E-mail addresses:* aurelien.naldi@gmail.com, naldi@tagc.univ-mrs.fr (A. Naldi).

0304-3975/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.10.021

reduction method allows the user to select candidate nodes for reduction and to perform dynamical analyses on reduced model versions that preserve relevant topological and dynamical properties.

The paper is organised as follows. Section 2 recalls the definitions of logical regulatory graphs and of their associated state transition graphs. Next, the reduction method is defined in Section 3 and its implementation is presented in Section 4. Relationships between the dynamical behaviour of the original model and that of the reduced model are delineated in Section 5. A multi-valued logical model of the segment polarity network is then used to demonstrate the power of the proposed reduction method in Section 6. The paper ends with conclusions and further prospects.

All models presented in this paper can be opened, edited, simulated, and analysed with *GINsim*.

## 2. Logical modelling of regulatory networks

Our modelling approach relies on the generalised logical formalism initially developed by Thomas et al. [5,6,3]. In this context, a regulatory network and its dynamics are both represented in terms of oriented graphs.

### 2.1. Logical regulatory graphs

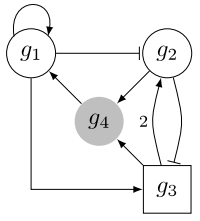
**Definition 1.** *A logical regulatory graph (LRG)* is a directed labelled multigraph $\mathcal{R} = (\mathcal{G}, \mathcal{M}ax, \Gamma, \Theta, \mathcal{K})$ where,

- $\mathcal{G} = \{g_1, \ldots, g_N\}$ is the set of nodes, representing *regulatory components*.
- $\mathcal{M}ax : \mathcal{G} \to \mathbb{N}^*$ associates a *maximum level* $\mathcal{M}ax(g_i) = \mathcal{M}ax_i$ to node $g_i$. The *current level* of $g_i$, denoted $x_i$, takes its values in $\mathcal{D}_i = \{0, \ldots, \mathcal{M}ax_i\}$.
- $\Gamma$ is the set of arcs, defined as a finite multiset of ordered pairs of elements of $\mathcal{G}$ representing *regulatory interactions*. If $\mathcal{M}ax_i > 1$, $g_i$ may have different effects on 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 $m_{i,j}$ $(1 \leq m_{i,j} \leq \mathcal{M}ax_i)$. Loops (even multi-loops) are allowed: an arc $(g_i, g_i)$ denotes an autoregulation of $g_i$.

  For each $g_j \in \mathcal{G}$, $Reg(j)$ denotes the set of its regulators: $g_i \in Reg(j)$ if and only if $(g_i, g_j) \in \Gamma$.
- $\Theta$ is a labelling function, which associates a *threshold* with each element of $\Gamma$. More precisely, $\theta_{i,j,k}$ is associated with the $k$th interaction between $g_i$ and $g_j$ (denoted $(g_i, g_j, \theta_{i,j,k})$, $k \in \{1, \ldots, m_{i,j}\}$), with $1 \leq \theta_{i,j,1} < \cdots < \theta_{i,j,m_{i,j}} \leq \mathcal{M}ax_i$. This interaction is *active*, when $x_i$, the level of its source $g_i$, lies 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, $\theta_{i,j,m_{i,j}+1} = \mathcal{M}ax_i + 1$).
- $\mathcal{K} = (\mathcal{K}_1, \ldots, \mathcal{K}_N)$ 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 $g_i$:

$$\mathcal{K}_i : \left( \prod_{g_j \in \mathcal{G}} \mathcal{D}_j \right) \mapsto \{0, \ldots, \mathcal{M}ax_i\}.$$

Fig. 1 illustrates this definition of a logical regulatory graph. In the following, when no confusion is possible, we will use $i$ to denote $g_i$.




$$\mathcal{G} = \{g_1, g_2, g_3, g_4\}$$
$$\mathcal{M}ax_1 = \mathcal{M}ax_2 = \mathcal{M}ax_4 = 1$$
$$\mathcal{M}ax_3 = 2$$
$$\mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_4 = \{0, 1\}, \mathcal{D}_3 = \{0, 1, 2\}$$
$$Reg(2) = \{g_1, g_3\}$$
$$m_{3,2} = 1$$
$$\theta_{3,2,1} = 2$$

**Fig. 1.** Example of a logical regulatory graph. Left: graphical representation of a LRG. Blunt arrows depict inhibitions while normal arrows depict activations (this is only a graphical convention since the regulatory effects are defined by the logical functions). The rectangular node $g_3$ is ternary, whereas the other nodes are Boolean. The thresholds of all interactions are set to 1, except that of $(g_3, g_2)$, which is set to 2. Right: illustration of the notations of Definition 1. Examples of logical functions $\mathcal{K}_i$ are displayed in Fig. 2 for the same model.

### 2.2. State transition graphs

We represent the dynamical behaviour of a LRG in the form of a state transition graph, defined as follows.

**Definition 2.** Given a LRG $\mathcal{R} = (\mathcal{G}, \mathcal{M}ax, \Gamma, \Theta, \mathcal{K})$, its associated *full state transition graph* (STG) $\mathcal{E} = (\mathcal{S}, \mathcal{T})$ is a directed graph, where:

- $\mathcal{S} = \Pi_{i\in\mathcal{G}}\mathcal{D}_i$ is the state space, a state of the system being a vector $x = (x_i)_{i=1,\ldots,N}$, with $x_i \in \mathcal{D}_i,\ \forall i \in \mathcal{G}$,
- $\mathcal{T} \subset \mathcal{S}^2$ is the set of transitions defined as follows: $(x, y) \in \mathcal{T}$ if and only if $\exists i \in \mathcal{G}$ such that:
  $x_i \neq \mathcal{K}_i(x)$,
  $y = x + \Delta_i(x).e^i$, where $\Delta_i(x) = \frac{\mathcal{K}_i(x)-x_i}{|\mathcal{K}_i(x)-x_i|}$ and $e^i$ is the canonical vector in $\mathcal{S}$ ($e^i_i = 1$ and $e^i_j = 0,\ \forall j \in \mathcal{G}, j \neq i$).

Here $\Delta_i(x)$ gives the direction of the update of $i$ (increase or decrease). One can also consider a state transition graph related to an initial (set of) condition(s). It is then a subgraph of the full STG.

When analysing the behaviour of a LRG, we mainly focus on attractors, which represent asymptotic dynamical properties. Given a STG, attractors are its terminal strongly connected components, classified as:

- *stable states*: reduced to a unique terminal node,
- *cyclic attractors*: terminal elementary (oriented) cycles,
- *complex attractors*: other terminal strongly connected components (*i.e.* involving intertwined cycles).

Cyclic and complex attractors will be called *non-trivial* attractors.

### 2.3. Effective interactions

According to Definition 1, regulatory interactions can be deduced from the logical functions $\mathcal{K}$. We say that an interaction $(i, j, \theta)$ is *effective* if there exists a state $x \in \mathcal{S}$ such that: $x_i = \theta - 1$, and $\mathcal{K}_j(x) \neq \mathcal{K}_j(x + e^i)$.

In practice, regulatory graphs are often manually constructed, *e.g.* from interactions documented in the literature, first defining the sets $\Gamma$ and $\Theta$. Next, the behaviours of the components are specified (through the logical functions). As a consequence, a LRG may contain non-effective interactions. Obviously, non-effective interactions can be safely discarded from the model with no consequence on the resulting dynamics.

Consider now the case of autoregulated nodes. An effective autoregulation $(i, i, \theta)$ is said to be *functional* if the values of the logical function flank the threshold $\theta$ of the autoregulation, *i.e.* for $x \in \mathcal{S}$ with $x_i = \theta - 1$ and $x' = x + e^i$, we have: $\mathcal{K}_i(x) < \theta \leq \mathcal{K}_i(x')$ or $\mathcal{K}_i(x) \geq \theta > \mathcal{K}_i(x')$ (see [7]).

Note that, in the Boolean case, all effective autoregulations are also functional, but effective non-functional autoregulations may appear in multi-valued models. In this case, even if the target values are different ($\mathcal{K}_i(x) \neq \mathcal{K}_i(x')$), $x_i$ tends to the same direction. As a consequence, non-functional autoregulations do not affect the dynamical behaviour and can thus be removed along with non-effective interactions. The removal of such autoregulations requires to update the logical functions: if both $\mathcal{K}_i(x)$ and $\mathcal{K}_i(x')$ are smaller or equal to $x_i$, set $\mathcal{K}_i(x) = \mathcal{K}_i(x')$, otherwise (both are greater than $x_i$), set $\mathcal{K}_i(x') = \mathcal{K}_i(x)$. After these changes, all remaining effective autoregulations are functional.

Hereafter, we consider only LRGs devoid of non-effective interactions and of non-functional autoregulations.

## 3. Logical regulatory graph reduction

This section presents the principles underlying the reduction of a regulatory graph and defines the resulting model, called *reduced model*. In what follows, we consider a reduction consisting of the *removal* of a single regulatory component. The generalisation to a reduction encompassing a set of nodes is obtained by iterating single node reductions. However, the ordering of a sequence of one-node reductions may have an impact on the resulting reduced model (see further discussion in Section 7).

We aim at defining a reduction method that preserves the salient dynamical properties of the original model. The underlying principle is already intuitively applied by modellers when they make regulatory nodes implicit in their networks. Removing a node basically consists in connecting directly its regulators to its targets. This is done by browsing the possible values of its regulators and by considering that the removed component is stable (*i.e.* at its target value). In other words, we consider the update of the value of the removed component as a *fast process*, which is performed before anything else.

Following this principle, it is impossible to remove an autoregulated component since it would not have a unique target value for fixed values of its other regulators. Thus, the removal of an autoregulated component implies additional decisions, impeding the definition of a systematic procedure. In the following, we will require that autoregulated components should not be removed.

To properly implement an algorithm producing the reduced model, we need further notations to manipulate the logical functions. Given a regulatory graph $\mathcal{R} = (\mathcal{G}, \mathcal{M}ax, \Gamma, \Theta, \mathcal{K})$ and a node $i \in \mathcal{G}$, we denote:

- $x_i^{\{l\}}$ ($l \in \mathcal{D}_i$) the Boolean variable with value 1 when $x_i = l$, 0 otherwise.
- $x_i^S$ the Boolean variable that is true if $x_i \in S$, false otherwise. Hence $x_i^S$ is defined by,

$$x_i^S \triangleq \bigvee_{l\in S} x_i^{\{l\}}, \quad S \subseteq \mathcal{D}_i.$$

Note that $x_i^{\emptyset}$ is always false and $x_i^{\mathcal{D}_i}$ always true.

- For all $v \in \mathcal{D}_i$, the logical function $\mathcal{K}_i^v$ that gives the conditions under which the target value of node $i$ is $v$. This function is defined as follows:

$$\mathcal{K}_i^v = \bigvee_{n=1,\dots p} \mathcal{C}_i^n, \tag{1}$$

where $\mathcal{C}_i^n$ are conjunctive clauses $\mathcal{C}_i^n = \bigwedge_{j \in Reg(i)} x_j^{S_{j,i,n}}$, with $S_{j,i,n} \subseteq \mathcal{D}_j$. Each clause $\mathcal{C}_i^n$ defines a situation (*i.e.* sets of combinations of incoming interactions acting upon $i$) for which the target value of $i$ is $v$.

In Eq. (1), each clause $\mathcal{C}_i^n$ defines a subset of $\mathcal{S}$, $D = \Pi_{j \in \mathcal{G}} S_{j,i,n}$ (with $S_{j,i,n} = \mathcal{D}_j$, $\forall j \notin Reg(i)$), such that for all $x \in D$, $\mathcal{K}_i(x) = v$. Hence, Eq. (1) defines a set of cubes in the state space $\mathcal{S}$, where the target value of $i$ is $v$.

**Definition 3.** Given a LRG $\mathcal{R} = (\mathcal{G}, \mathcal{M}ax, \Gamma, \Theta, \mathcal{K})$, the *reduced LRG* $\mathcal{R}^r = (\mathcal{G}^r, \mathcal{M}ax^r, \Gamma^r, \Theta^r, \mathcal{K}^r)$ obtained by removing a non-autoregulated component $r \in \mathcal{G}$ is defined as follows:

- $\mathcal{G}^r = \mathcal{G} \setminus \{r\}$.
- $\mathcal{M}ax^r : \mathcal{G}^r \to \mathbb{N}^*$, s.t. $\forall i \in \mathcal{G}^r\ \mathcal{M}ax^r(i) = \mathcal{M}ax_i$.
- For all $i \in \mathcal{G}^r$, and for all $v \in \mathcal{D}_i$, the logical function $\mathcal{K}_i^{rv}$ is defined as follows. Consider $\mathcal{K}_i^v = \bigvee_{n=1,\dots p} \mathcal{C}_i^n$, the disjunctive form of $\mathcal{K}_i^v$, as defined previously. For all $n = 1, \dots p$ (*i.e.* for each clause $\mathcal{C}_i^n$), $\mathcal{F}_i^{rn}$ is defined as:

$$\mathcal{F}_i^{rn} = \left( \bigvee_{w \in S_{r,i,n}} \mathcal{K}_r^w \right) \wedge \left( \bigwedge_{j \in Reg(i) \setminus \{r\}} x_j^{S_{j,i,n}} \right).$$

Then $\mathcal{K}_i^{rv} = \bigvee_{n=1,\dots p} \mathcal{F}_i^{rn}$.
- $\Gamma^r$ and $\Theta^r$ are deduced from $\mathcal{K}^r$;

$$\forall i \in \mathcal{G}^r,\ j \in \mathcal{G}^r \quad m_{i,j}^r = \sum_{v \in [1, \mathcal{M}ax_i]} \mathbb{1}_{i,j,v},$$

where $\mathbb{1}_{i,j,v} = 1$ if there exists $x \in \mathcal{S}$ such that $x_i = v - 1$ and $\mathcal{K}_j^r(x) \neq \mathcal{K}_j^r(x + e^i)$, $\mathbb{1}_{i,j,v} = 0$ otherwise. Then $(i,j) \in \Gamma^r$ if $m_{i,j}^r > 0$ (and the multiplicity of $(i,j)$ in $\Gamma^r$ is given by $m_{i,j}^r$). Finally, the ordered set of values $v$ such that $\mathbb{1}_{i,j,v} = 1$ defines the thresholds $\theta_{i,j,k}^r$ $(k = 1, \dots, m_{i,j}^r)$.

The logical function $\mathcal{K}_i^{rv}$ is deduced from the logical function $\mathcal{K}_i^v$ by replacing, in each clause, literals $x_r^S$ by the formulae giving the conditions under which the target value of $r$ is in $S$ (note that this definition may not give $\mathcal{K}_i^{rv}$ in a proper disjunctive form). If $\mathcal{C}_i^n$ does not depend on $r$ (*i.e.* $r \notin Reg(i)$) then $S_{r,i,n} = \mathcal{D}_r$ and $\mathcal{F}_i^{rn} = \mathcal{C}_i^n$ for all $n$, therefore $\mathcal{K}_i^{rv} = \mathcal{K}_i^v$. Note that in defining $\Theta^r$ and $\Gamma^r$ from the logical functions, all interactions correspond to *effective* interactions as defined in Section 2.3, but there might be non-functional autoregulations (on multi-valued targets of $r$ in $\mathcal{R}$). These can be removed as explained in Section 2.3.

The set of arcs verifies:

$$\Gamma^r \subseteq \{(i,j) \in \mathcal{G}^r \times \mathcal{G}^r,\ \text{s.t.}\ (i,r),\ (r,j) \in \Gamma \text{ or } (i,j) \in \Gamma\}.$$

## 4. Implementation

In practice, the construction of the new logical functions is performed using Reduced Ordered Multivalued Decision Diagrams (ROMDDs or MDDs for short). Decision diagrams are rooted directed acyclic graphs, widely used to represent logical functions (see *e.g.* [8,9]). In these diagrams, internal nodes are labelled with *decision variables* and have one child per value, while leaves represent the values of the function. Decision variables are ordered: each internal node has a rank and the sub-diagrams rooted by the children of a node of rank $i$ do not contain internal nodes of rank $j \leq i$. In [7], we used MDDs to represent the logical functions $\mathcal{K}_i$. In this context, decision variables denote the levels of the components of the model. For the sake of simplicity, we consider that the ordering of the MDD variables is the same as that of the LRG components. Given the MDD representation of $\mathcal{K}_i$ and a state $x$, a unique path from the root of the MDD to one of its leaves 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 $\mathcal{K}_i(x)$. Each clause of $\mathcal{K}_i^v$ corresponds to a path leading to a leaf valued $v$.

To compute the MDD representing $\mathcal{K}_i^r$, we define a recursive algorithm, taking as input the MDDs representing $\mathcal{K}_i$ (denoted $\mathcal{K}$ in the algorithm) and $\mathcal{K}_r$. The principle is to scan these two MDDs at the same time until a node of rank $r$ is encountered in $\mathcal{K}$. All its children are then considered until the value of $r$ is determined by reaching a leaf in $\mathcal{K}_r$. The branch of $\mathcal{K}$ corresponding to the value labelling this leaf is then retained and the path from the root of the MDD to the root of this sub-diagram is reconstructed as the recursion unwinds.

MDD leaves have a value (denoted $\mathcal{K}$.value below), while internal MDD nodes have a rank ($\mathcal{K}$.rank) and an array of $Max_{\mathcal{K}.rank} + 1$ children ($\mathcal{K}$.child, where $\mathcal{K}$.child[i] is the $i$th child, corresponding to $x_{\mathcal{K}.rank} = i$). The function CREATE(p, child[]) creates an internal node of rank p with the children specified in the array child[] and takes care of the usual MDD simplifications. MDD representations of logical functions and the outcome of this algorithm are illustrated in Fig. 2.

```
// INPUTS:
// − K: MDD associated to the component
// − r: index of the removed component
// − K_r: MDD associated to r
// OUTPUT: modified MDD for the component
REMOVE(K, r, K_r):
  if K is a leaf or K.rank > r:
    return K   //r has no effect
  if K.rank = r:
    if K_r.rank = r:
      ERROR: r is autoregulated
    else: //r found, browse K_r and K's children
      return FIX(K.child[],K_r)

  // recursively build child[] using K, K_r or
  // their children, depending on their rank
  p ← minimum of K_r.rank, K.rank
  for v ← 0 to Max_p:
    if K.rank = K_r.rank:
      child[v] ← REMOVE(K.child[v],r,K_r.child[v])
    else if K.rank > K_r.rank:
      child[v] ← REMOVE(K, r, K_r.child[v])
    else:
      child[v] ← REMOVE(K.child[v],r,K_r)
  return CREATE(p,child[])


//INPUTS:
// − mdd[]: array of the sub−MDDs of the component,
//          for each value of r
// − K_r: sub−MDD associated to r
//OUTPUT: new sub−MDD for the component
FIX(mdd[],K_r):
  if K_r is a leaf: // fixed value of r
    return mdd[K_r.value]

  // recursively build the children, with a new
  // child[] filled according to the ranks in mdd[]
  p ← minimum of K_r.rank, K.rank, ∀ K ∈ mdd[]
  for v ← 0 to Max_p:
    for i ← 0 to len(mdd)−1:
      if mdd[i].rank = p:  //consider the v{th} child
        t_mdd[i] ← mdd[i].child[v]
      else:  //consider the current MDD
        t_mdd[i] ← mdd[i]
    if p = K_r.rank:
      child[v] ← FIX(t_mdd[],K_r.child[v])
    else:
      child[v] ← FIX(t_mdd[],K_r)
  return CREATE(p,child[])
```

## 5. Dynamics of the reduced model

In this section, the dynamical behaviour of a reduced LRG (as specified in Definition 3) is compared to that of the original LRG. In particular, we show that the reduction preserves existing attractors and does not add any spurious path.

Let $\mathcal{E} = (\mathcal{S}, \mathcal{T})$ be the full state transition graph of $\mathcal{R} = (\mathcal{G}, \mathcal{M}ax, \Gamma, \Theta, \mathcal{K})$ and $r \in \mathcal{G}$ a node not autoregulated. Let $\mathcal{E}^r = (\mathcal{S}^r, \mathcal{T}^r)$ be the full STG of $\mathcal{R}^r = (\mathcal{G}^r, \mathcal{M}ax^r, \Gamma^r, \Theta^r, \mathcal{K}^r)$, the LRG obtained after the removal of $r$ from $\mathcal{G}$.

Consider the projection $\pi_r : \mathcal{S} \to \mathcal{S}^r$ such that, $\forall i \in \mathcal{G}^r$, $\forall x \in \mathcal{S}$, $(\pi_r(x))_i = x_i$, and the equivalence relation on $\mathcal{S}$: $\forall x, y \in \mathcal{S}$, $x \sim_r y$ iff $\pi_r(x) = \pi_r(y)$.

We denote $[x]_{\sim r}$ the equivalence class: $[x]_{\sim r} = \{y \in \mathcal{S} \text{ s.t. } y \sim_r x\}$. The class $[x]_{\sim r}$ contains all states of $\mathcal{S}$ that differ only by their $r$th component, *i.e.* the $(\mathcal{M}ax_r + 1)$ states $\{x^i \in \mathcal{S},\ i = 0, \ldots, \mathcal{M}ax_r\}$, such that $x^i \sim_r x$ and $x^i_r = i$. Because $r$ is not autoregulated, $\forall x^i \in [x]_{\sim r}$, $\mathcal{K}_r(x^i) = \mathcal{K}_r(x)$. This implies that:

- $(x^i, x^{i+1}) \in \mathcal{T}$, for all $0 \leq i < \mathcal{K}_r(x)$,

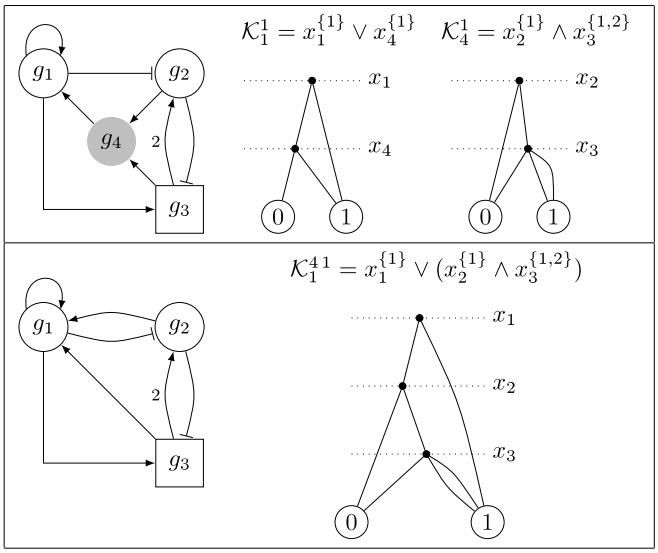



**Fig. 2.** Reduction in terms of MDDs. Top: the same LRG as in Fig. 1, where $g_4$ (greyed-out) is selected for removal. Logical functions for $g_1$ and $g_4$ are shown on the right, along with their MDD representations. Bottom: the reduced LRG after removal of $g_4$, along with the resulting logical function for $g_1$. In the MDDs, internal nodes are labelled with the associated variable ($x_i$), whereas leaves represent the value of the logical functions. Children of internal nodes are ordered from left to right: the leftmost (resp. rightmost) child is the root of the sub-diagram corresponding to the case $x_i = 0$ (resp. $x_i = \mathit{Max}_i$).

- $(x^i, x^{i-1}) \in \mathcal{T}$, for all $\mathcal{K}_r(x) < i \leq \mathit{Max}_r$,
- $(x^{\mathcal{K}_r(x)}, x^i) \notin \mathcal{T}$, for all $0 \leq i \leq \mathit{Max}_r$.

Hence, for all $x \in \mathcal{S}$, there exists a path in $\mathcal{S}$ from $x$ to $x^{\mathcal{K}_r(x)}$ that is the representative state of $[x]_{\sim_r}$.

**Definition 4.** A state $x \in \mathcal{S}$ is the *representative state* of an equivalence class for $\sim_r$ iff $x_r = \mathcal{K}_r(x)$ (with $r$ a non-autoregulated component).

Note that when $r$ is not autoregulated, the representative state does exist and is unique. We can then define the *retrieval* function $s_r : \mathcal{S}^r \to \mathcal{S}$ such that, $\forall z \in \mathcal{S}^r$,

$$(s_r(z))_i = z_i, \quad \text{for all } i \in \mathcal{G} \setminus \{r\},$$
$$(s_r(z))_r = \mathcal{K}_r(x), \quad \text{with } x \text{ such that } \pi_r(x) = z.$$

The state $s_r(z)$ is the representative state of the equivalence class whose states are projected on $z$ (see Fig. 3). On this basis, we can introduce an alternative definition of the logical functions in the reduced LRG: $\forall i \in \mathcal{G}^r$, $\mathcal{K}_i^r : \mathcal{S}^r \mapsto \mathcal{D}_i$ is defined as $\mathcal{K}_i^r(z) = \mathcal{K}_i(s_r(z))$. Note that if $(r, i) \notin \Gamma$ (*i.e.* $r$ is not a regulator of $i$), $\mathcal{K}_i^r(\pi_r(x)) = \mathcal{K}_i(x)$.

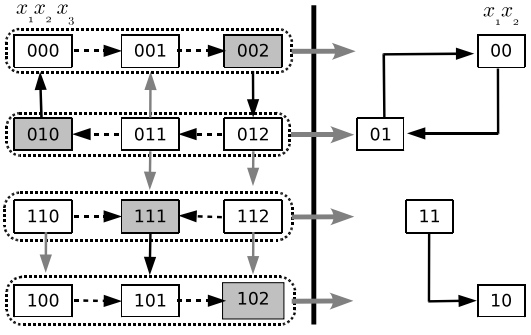



**Fig. 3.** Dynamical behaviour of the reduced model given in Fig. 2 bottom, before and after the removal of the ternary node $g_3$. Left: State transition graph (STG), partitioned into four equivalence classes. Each equivalence class contains 3 states differing by the value of $g_3$; its representative state is greyed out and internal transitions are dashed. Right: STG of the reduced model, each state corresponding to an equivalence class of the original STG. After the reduction, the stable state 102 is projected on 10 and all transitions are preserved except the one from the second equivalence class to the third one. This results in the isolation of the non-terminal strongly connected component involving states of the first two equivalence classes of the original STG, hence generating the attractor (01, 00).

**Remark 1.** It follows from their definitions that functions $\pi_r$ and $s_r$ verify:

1. $\pi_r \circ s_r$ is the identity function.
2. For any $x \in \mathcal{S}$, $(s_r \circ \pi_r(x)) \sim_r x$.

3. If $x \in \mathcal{S}$ is a representative state, then $s_r \circ \pi_r(x) = x$.
4. For any $z \in \mathcal{S}^r$, $\mathcal{K}^r(z) = \pi_r(\mathcal{K}(s_r(z)))$. Indeed, $\forall x \in \mathcal{S},\ \forall i \in \mathcal{G}^r$, $\mathcal{K}^r_i(\pi_r(x)) = \mathcal{K}_i(s_r \circ \pi_r(x))$.

The following lemma establishes the relationships between transitions in $\mathcal{E}$ and $\mathcal{E}^r$.

**Lemma 1.** 1. *Let* $z, z' \in \mathcal{S}^r$.

$$(z, z') \in \mathcal{T}^r \Longrightarrow \exists x \in \mathcal{S} \text{ s.t. } \pi_r(x) = z' \quad \text{and} \quad (s_r(z), x) \in \mathcal{T}.$$

2. *Let* $x,\ y \in \mathcal{S}$. *If* $x$ *is a representative state, then*

$$(x, y) \in \mathcal{T} \Longrightarrow (\pi_r(x), \pi_r(y)) \in \mathcal{T}^r.$$

**Proof.** Recall the notation $\Delta_i(x) = \frac{\mathcal{K}_i(x) - x_i}{|\mathcal{K}_i(x) - x_i|}$, with $x \in \mathcal{S}$ such that $\mathcal{K}_i(x) \neq x_i$. For $z \in \mathcal{S}^r$ s.t. $z_i \neq \mathcal{K}^r_i(z)$, we similarly denote:

$$\Delta^r_i(z) \triangleq \frac{\mathcal{K}^r_i(z) - z_i}{|\mathcal{K}^r_i(z) - z_i|} = \frac{\mathcal{K}_i(s_r(z)) - (s_r(z))_i}{|\mathcal{K}_i(s_r(z)) - (s_r(z))_i|} = \Delta_i(s_r(z)).$$

1. Consider $z, z' \in \mathcal{S}^r$ such that $(z, z') \in \mathcal{T}^r$. Then $\exists i \neq r$ s.t. $\mathcal{K}^r_i(z) \neq z_i$, and $z' = z + \Delta^r_i(z)\, e^i$. By definition, $\mathcal{K}^r_i(z) = \mathcal{K}_i(s_r(z)) \neq (s_r(z))_i = z_i$. This implies that $(s_r(z), x) \in \mathcal{T}$ with $x \in \mathcal{S}$ and $x = s_r(z) + \Delta_i(s_r(z))\, e^i$, and then $\pi_r(x) = z'$.
2. Consider $x, y \in \mathcal{S}$ such that $\mathcal{K}_r(x) = x_r$. The hypothesis $(x, y) \in \mathcal{T}$ implies that $\exists i \in \mathcal{G}, i \neq r$ s.t. $\mathcal{K}_i(x) \neq x_i$, and $y = x + \Delta_i(x)\, e^i$.

   We have $\mathcal{K}^r_i(\pi_r(x)) = \mathcal{K}_i(x)$ (since $x$ is a representative state), and $x_i = (\pi_r(x))_i$, since $i \neq r$. So, $\mathcal{K}^r_i(\pi_r(x)) \neq (\pi_r(x))_i$, and then $\exists z \in \mathcal{S}^r$ s.t. $(\pi_r(x), z) \in \mathcal{T}^r$, with

$$z = \pi_r(x) + \Delta^r_i(\pi_r(x))\, e^i = \pi_r(x) + \Delta_i(s_r \circ \pi_r(x))\, e^i = \pi_r(y). \quad \square$$

The first item of Lemma 1 states that any transition in $\mathcal{T}^r$ corresponds to at least one transition in $\mathcal{T}$. Clearly, the reverse is not true. The second item of the lemma gives a condition under which transitions are preserved from $\mathcal{T}$ to $\mathcal{T}^r$. Of course, it is important to know which transitions are lost through the reduction.

**Definition 5.** The reduction *preserves a transition* $(x, y) \in \mathcal{T}$ if $(\pi_r(x), \pi_r(y)) \in \mathcal{T}^r$, or $\pi_r(x) = \pi_r(y)$. The reduction *preserves a path* $(s_1, \ldots, s_n) \in \mathcal{E}$ if all its transitions are preserved.

In other words, a path $(s_1, \ldots, s_n)$ in $\mathcal{E}$ is preserved if the reduction preserves the transitions between equivalence classes, in the required order.

The following property characterises the transitions that are not preserved by the reduction.

**Property 1.** *A transition* $(x, y) \in \mathcal{T}$ *is* not *preserved by the reduction if and only if the three following conditions are satisfied:*

1. $x$ *is not a representative state,*
2. $y \notin [x]_{\sim_r}$ $(\Rightarrow\ \exists i \neq r \text{ s.t. } y_i \neq x_i)$,
3. $\Delta_i(x) \neq \Delta_i(s_r \circ \pi_r(x))$ .

The last condition means that there is no call for updating $i$ in the same direction in state $s_r \circ \pi_r(x)$.

**Proof.** Consider a transition $(x, y) \in \mathcal{T}$, which satisfies the three conditions. Suppose that $(x, y)$ is preserved by the reduction, then $(\pi_r(x), \pi_r(y)) \in \mathcal{T}^r$ (the case $\pi_r(x) = \pi_r(y)$ is not possible because of the second condition). This means that there exists $j \neq r$ s.t. $(\pi_r(x))_j \neq (\pi_r(y))_j$, and $(\pi_r(x))_k = (\pi_r(y))_k$ for any $k \neq j$. With Condition 2 and by definition of $\pi_r$, we deduce that $j = i$. Moreover, we know that:

$$\pi_r(y) = \pi_r(x) + \Delta^r_i(\pi_r(x))\, e^i = \pi_r(x) + \Delta_i(s_r \circ \pi_r(x))\, e^i.$$

Finally, $y = x + \Delta_i(x)\, e^i$, and, as $y_i = (\pi_r(y))_i$, we have $\Delta_i(x) = \Delta_i(s_r \circ \pi_r(x))$. This contradicts Condition 3. Hence, $(x, y)$ is not preserved by the reduction.

Conversely, let $(x, y) \in \mathcal{T}$ be a transition not preserved by the reduction.

- Condition 1 is satisfied by the second item of Lemma 1.
- Condition 2 is satisfied because $y \in [x]_{\sim_r} \Rightarrow \pi_r(x) = \pi_r(y) \Rightarrow (x, y)$ preserved, hence a contradiction.
- We know that $y = x + \Delta_i(x)\, e^i$. As $\mathcal{K}^r(\pi_r(x)) = \pi_r(\mathcal{K}(s_r \circ \pi_r(x)))$ (cf. Remark 1),

$$\begin{aligned} \mathcal{K}^r_i(\pi_r(x)) &= (\pi_r(\mathcal{K}(s_r \circ \pi_r(x))))_i = \mathcal{K}_i(s_r \circ \pi_r(x)) \\ &\neq x_i = (\pi_r(x))_i. \end{aligned}$$

Hence, there exists $z \in \mathcal{S}^r$ s.t. $(\pi_r(x), z) \in \mathcal{T}^r$ with

$$\begin{aligned} z_i &= \pi_r(x) + \Delta^r_i(\pi_r(x))\, e^i \\ &= \pi_r(x) + \Delta_i(s_r \circ \pi_r(x))\, e^i = \pi_r(x) + \Delta_i(x)\, e^i. \end{aligned}$$

Consequently, $\pi_r(y) = z$ and $(x, y)$ is preserved, hence a contradiction. $\square$

Given $C$, a set of states in $\mathcal{S}$, we denote $\pi_r(C) \triangleq \{\pi_r(x),\ x \in C\}$. Given $C'$, a set of states in $\mathcal{S}^r$, we denote $s_r(C') \triangleq \{s_r(z),\ z \in C'\}$. Note that $\pi_r(C)$ may contain less elements than $C$, and that $s_r(C')$ contains only representative states. The following results relate attractors in $\mathcal{E}$ and $\mathcal{E}^r$.

**Theorem 1.** *Consider a LRG $\mathcal{R} = (\mathcal{G}, Max, \Gamma, \Theta, \mathcal{K})$ and $\mathcal{R}^r$ the reduced LRG. Let $\mathcal{E}$ (resp. $\mathcal{E}^r$) be the full STG of $\mathcal{R}$ (resp. of $\mathcal{R}^r$), then:*

1. *Stable states in $\mathcal{E}$ and $\mathcal{E}^r$ verify:*
   - *$x$ stable state in $\mathcal{E} \implies x$ is a representative state and $\pi_r(x)$ stable state in $\mathcal{E}^r$. Furthermore no other stable state is projected on $\pi_r(x)$,*
   - *$z$ stable state in $\mathcal{E}^r \implies s_r(z)$ stable state in $\mathcal{E}$.*

   *Hence, the number of stable states is conserved by the reduction.*
2. *If $(s_1, \ldots s_n)$ is a cyclic attractor in $\mathcal{E}$, then $(\pi_r(s_1), \ldots \pi_r(s_n))$ is a cyclic attractor in $\mathcal{E}^r$.*
3. *If $C$ is a complex attractor in $\mathcal{E}$, $z \in \pi_r(C)$ and $(z, z') \in \mathcal{T}^r$, then $z' \in \pi_r(C)$. As a consequence, $\pi_r(C)$ contains at least one non-trivial attractor in $\mathcal{E}^r$.*

**Proof.** In what follows, we use $\mathcal{K}(x) \in \mathcal{S}$ to denote $(\mathcal{K}_i(x))_{i\in\mathcal{G}}$.

1. Let $x \in \mathcal{S}$ be a stable state: $\forall i \in \mathcal{G}$, $\mathcal{K}_i(x) = x_i$, hence $x$ is a representative state. Then, $\mathcal{K}^r(\pi_r(x)) = \pi_r(\mathcal{K}(s \circ \pi_r(x))) = \pi_r(x)$. Hence, $\pi_r(x)$ is a stable state in $\mathcal{S}^r$.

   Let $z \in \mathcal{S}^r$ be a stable state. For all $i \in \mathcal{G}^r$, we have $\mathcal{K}^r_i(z) = z_i$. Consequently, for all $i \in \mathcal{G}^r$, $\mathcal{K}_i(s_r(z)) = (s_r(z))_i$, and, by definition of $s_r$, $\mathcal{K}_r(s_r(z)) = (s_r(z))_r$.
2. A cyclic attractor $(s_1, \ldots s_n)$ in $\mathcal{E}$ is an elementary cycle such that $\forall i \in \{1, \ldots, n\}$, $s_i$ has a unique successor $s_{i+1}$. If $s_i$ is a representative state, then $\pi_r(s_i)$ has a unique successor $\pi_r(s_{i+1})$ in $\mathcal{T}^r$. Otherwise, $s_i \sim_r s_{i+1}$ (as the successor is unique). Hence, the path $(\pi_r(s_1), \ldots \pi_r(s_n))$ exists in $\mathcal{T}^r$ and each $\pi_r(s_i)$ has a unique successor. Thus $(\pi_r(s_1), \ldots \pi_r(s_n))$ is a cyclic attractor in $\mathcal{G}^r$.
3. Given $C$, a complex attractor in $\mathcal{E}$. For all $z \in \mathcal{S}^r$, if $z \notin \pi_r(C)$ then $z$ is not reachable from any state of $\pi_r(C)$. Suppose that there exist $x \in C$ and $z \in \mathcal{S}^r$ such that $(\pi_r(x), z) \in \mathcal{T}^r$ and $z \notin \pi_r(C)$. Then there exists $y \in \mathcal{S}$ such that $\pi_r(y) = z$ and $(s \circ \pi_r(x), y) \in \mathcal{T}$ (by Lemma 1). Moreover, there exists a path from $x$ to $s \circ \pi_r(x)$ (Remark 1), so $s \circ \pi_r(x) \in C$ ($C$ is a terminal strongly connected component). This implies that $y \in C$, and $z = \pi_r(y) \in \pi_r(C)$, which is a contradiction. As a consequence, all trajectories starting in $\pi_r(C)$, projection of the complex attractor $C$, are trapped in $\pi_r(C)$. Thus $\pi_r(C)$ contains at least one non-trivial attractor, recalling that stable states are conserved, while $C$ is devoid of stable state. □

Theorem 1 characterises the dynamical properties conserved by the reduction. Going further, it is possible to identify the situations leading to the generation of additional non-trivial attractors. A non-trivial attractor in the reduced STG corresponds to a (part of a) strongly connected component of the original STG. This SCC is itself a non-trivial attractor or involves outgoing transitions all in conflict with transitions concerning the removed component. In other words, we can fully characterise the set of states in the original STG giving rise to a non-trivial attractor in the reduced dynamics. Interestingly, this set corresponds to transient oscillatory behaviour from which the system cannot escape provided that updates of the removed component are always faster than other concurrent changes.

**Theorem 2.** *Given a non-trivial attractor $C^r$ in $\mathcal{E}^r$,*

1. *$s_r(C^r) = \{s_r(z),\ z \in C^r\}$ is part of a strongly connected component $C$ in $\mathcal{E}$.*
2. *Let $C' = \{x \in \mathcal{E}, \text{s.t. } \pi_r(x) \in C^r\}$ ($= \cup_{x\in s_r(C^r)}[x]_{\sim_r}$), the set of states whose projections are in $C^r$. Suppose $(x, y) \in \mathcal{T}$, such that $x \in C'$ and $y \notin C'$, then $(x, y)$ is not preserved.*
3. *Suppose $(x, y) \in \mathcal{T}$, such that $x \in s_r(C^r)$ and $y \notin C' \cap C$, then $(x, y)$ is not preserved.*

**Proof.** Consider $C^r$ a non-trivial attractor in $\mathcal{E}^r$, then $C^r$ is a strongly connected component.

1. Let $x, y \in s_r(C^r)$; then $x = s_r(z)$ and $y = s_r(z')$, $z, z' \in C^r$. Let $z^0 = z, z^1, z^2, \ldots, z^l = z'$ a path from $z$ to $z'$. For all $i \in \{1, \ldots l-1\}$, $(z^i, z^{i+1}) \in \mathcal{T}^r \Rightarrow \exists u \in \mathcal{S}$ such that $(s_r(z^i), u) \in \mathcal{T}$ and $\pi_r(u) = z^{i+1}$. Moreover, we know that there exists a path from $u$ to $s_r \circ \pi_r(u) = s_r(z^{i+1})$. Hence, there is a path from $x = s_r(z)$ to $y$.
2. Let $(x, y) \in \mathcal{T}$, such that $x \in C'$ and $y \notin C'$. Suppose that $(x, y)$ is preserved, then either $\pi_r(x) = \pi_r(y)$ or $(\pi_r(x), \pi_r(y)) \in \mathcal{T}^r$. Both cases imply $\pi_r(y) \in C^r$, thus $y \in C'$, which is a contradiction.
3. Let $(x, y) \in \mathcal{T}$, such that $x \in s_r(C^r)$ and $y \notin C' \cap C$. Suppose that $(x, y)$ is preserved, then either $\pi_r(x) = \pi_r(y)$ or $(\pi_r(x), \pi_r(y)) \in \mathcal{T}^r$.
   - $\pi_r(x) = \pi_r(y)$ is not possible because it would imply that $y \sim_r x$ and since $x$ is a representative state ($x \in_r (C^r)$), it is stable for $r$ and cannot be the source of a transition leading to a state in its equivalence class.
   - if $(\pi_r(x), \pi_r(y)) \in \mathcal{T}^r$, thus $\pi_r(y) \in C^r$ and $y \in C'$. Furthermore, $s_r \circ \pi_r(y) \in s_r(C^r) \subset C$. A path exists from $x \in C$ to $y$ (since $(x, y) \in \mathcal{T}$) and from $y$ to $s_r \circ \pi_r(y) \in C$ (by item 2 of Remark 1), thus $y \in C$, which is a contradiction. □

## 6. Application to the analysis of the dynamics of a logical model for Drosophila segment polarity network

We demonstrate the power and flexibility of our reduction method through its application to the segment polarity network, which plays a key role in the segmentation of the fly embryo. This system has been thoroughly analysed by developmental geneticists and has been already modelled using continuous [10–12] and logical approaches [13–15]. However, all these studies involved important simplifications of the network, particularly so as a proper modelling of its behaviour requires the chaining of several identical networks to account for inter-cellular interactions through Wingless (Wg) and Hedgehog (Hh) signalling. Describing the most complete model to date, [15] had to discard various components known to play important roles in Wg and Hh signalling to keep dynamical simulations and analyses computationally tractable for up to six cells. Based on this work, we propose a detailed logical model of the segment polarity network, taking into account six additional components (Arm, Cos2, Fu, SuFu, Smo and Zw3). The resulting regulatory graph encompasses 18 components (among which 10 are associated with Boolean variables and 8 to ternary variables) and 31 regulatory interactions (left part of Fig. 4).

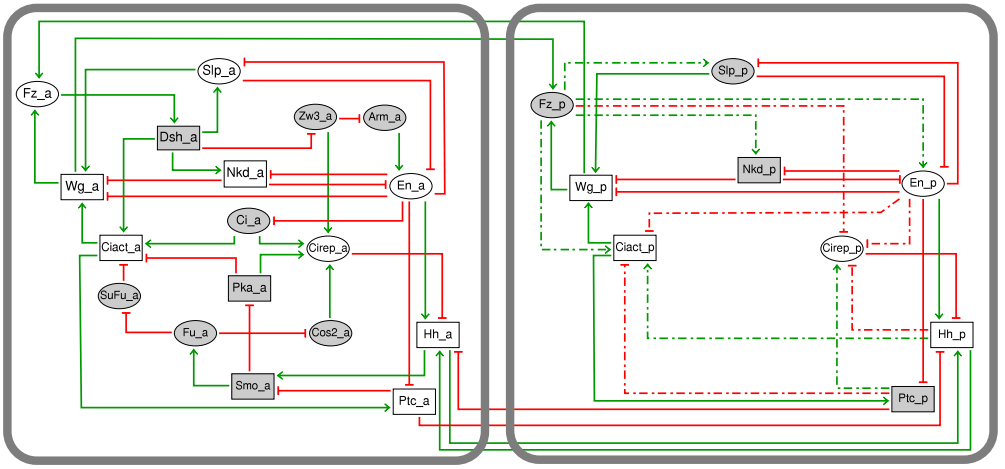



**Fig. 4.** Logical model of the segment polarity network for two cells, based on [15]. Ellipsoid and rectangular nodes denote Boolean and ternary components, respectively. The two cellular networks have been properly connected to take into account Wg and Hh diffusion, as well as Hh sequestration by Ptc, as in [16]. This figure illustrates a partially reduced model: the anterior cell contains the extended version of the model, while the right part has been reduced. Dashed arrows denote indirect interactions resulting from this reduction. Greyed-out components in the anterior cell are marked for reduction (and reduced in the posterior cell), while greyed-out components in the posterior cell are candidates for further reduction.

In order to model the intercellular interactions involved in the formation of segment boundaries, we connect neighbouring cells (along the anterior-posterior axis) through Wg and Hh signalling. Wg is known to bind its receptor, Frizzled (Fz), only at very short range, amounting here to neighbouring cells. This can be represented by positive arcs linking each Wg node to Fz nodes of neighbouring cells. In contrast, Hh is able to reach more distant cells, but can be sequestered by its receptor Patched (Ptc). Similar interactions have been modelled in [16] in terms of positive arcs between Hh nodes in neighbouring cells (diffusion) and negative arcs from Ptc onto the Hh node of neighbouring cells (sequestering). The resulting intercellular model is then reduced before performing analysis (Fig. 4 illustrates the intercellular network after reducing only one of two connected cells).

The reduction method described above can be advantageously applied to ease the identification of all attractors of such intercellular models (Sánchez et al. considered six cells [15]). The modeller can select the sets of nodes to discard from the network, depending on biological considerations (*e.g.* different time scales, specific mutations, etc.). In a first step, it is reasonable to conserve transcription factors and components involved in intercellular communications: Wg, Hh and their receptors (Fz and Ptc). However, since the transcription factor Cubitus interruptus is represented by three nodes here (full length immature Ci protein, activator Ci-act and repressor Ci-rep forms), we choose to retain only the two nodes corresponding to active regulatory forms. These choices correspond to the removal of the greyed-out components in the left part of Fig. 4.

The reduced model involves half of the nodes of the original one, which amounts to a much higher reduction of the number of possible states, as this grows exponentially with the number of regulatory nodes. The resulting regulatory graph (Fig. 4, right) remains easy to grasp as it reasonably unfolds the intra-cellular and inter-cellular regulatory pathways. As we shall see, this logical model can be further reduced to facilitate analyses encompassing more cells.

For proper logical rules (cf. [15] and supplementary material), one can check that the detailed and the reduced two-cells models have exactly the same number of stables states (as predicted by Theorem 1). These multi-cellular stable states combine three types of cellular states: a Wg expressing state (denoted W), an En expressing state (E), and a *trivial* state (T) expressing neither Wg, nor En. The three stable states found for the two connected cells correspond to the TT, WE and EW cell combinations reported by Sánchez et al. [15]. All three stable states are reachable from biologically relevant initial

**Table 1**
Dynamical characteristics of different reduced models derived from that of Fig. 4 (involving $2 \times 9$ nodes after applying the same reduction to both cells). The number of reachable states decreases drastically with the number of considered nodes. Note that the three stable states remain reachable for all reductions listed apart from the last one (removal of Slp).

| LRG size | Removed components | # reached states | Reached stable states |
|---|---|---|---|
| $2 \times 9$ | – | $> 10^6$ | TT, WE, EW |
| $2 \times 7$ | Fz, Ptc | 12 476 | TT, WE, EW |
| $2 \times 6$ | Fz, Ptc, Nkd | 1625 | TT, WE, EW |
| $2 \times 8$ | Slp | 11 350 | TT, WE |

conditions (significant amounts of Wg and Slp in the anterior cell, significant amounts of En in the posterior cell), provided as an outcome of the activity of the pair-rule system, cf. [17,15]. However, the size of the corresponding state transition graph still impedes detailed dynamical analyses (see Table 1).

As shown in Table 1, the removal of Fz, Ptc and Nkd drastically reduces the number of reached states without changing the reachability of the three stable states from the considered initial state. However, the sole removal of Slp impedes the reachability of the stable state with inverted Wg and En expressing cells. It also suppresses the functionality contexts of all negative circuits [7,15], implying that the state transition graph does not contain any cyclic attractor. Indeed, after further reduction to three nodes per cell (Wg, En and Hh), we were able to check the absence of non-trivial attractors in the full STG. As the reduction cannot delete existing non-trivial attractors (see Theorem 1), this implies that all attractors of the original model are stable states. This could not have been checked on the original full STG, which contains $2^{20} \times 3^{16} \simeq 45 \times 10^{12}$ states.

## 7. Discussion and prospects

We have defined a reduction method that can be applied to multi-valued logical regulatory graphs while preserving important dynamical properties. In particular, all attractors of the original dynamics have a counterpart in the dynamics of the reduced model. Furthermore, trajectories in the reduced model can be formally related to trajectories in the original one. This enables one to infer the existence of paths in the dynamics of a detailed model whenever it is possible to show (by simulation and graph analysis) that paths exist between the corresponding states in a reduced version of the model. However, the reverse is not true. Indeed, a reduction can lead to the loss of reachability properties. Whenever several components are called to asynchronously update their values in a given state, the reduction of one of these components amounts to considering that it is "faster" than the other ones. Indeed, the logical functions are modified by the reduction considering the target level of the reduced component in any given state. This leads to the possible exclusion of some transitions in the reduced STG, as established by Property 1. The resulting dynamical behaviour can be related to that obtained by applying specific priority classes [18]. Yet, the precise characterisation of this relationship requires further work.

Reduction strategies have long been considered to alleviate model analyses of large discrete event systems, from homomorphisms and bisimulations of transitions systems (*e.g.* [19]) to reduction techniques for Petri nets (*e.g.* [20–22]). These reductions also amount to lowering the size of the model (*i.e.* number of places or transitions), yet preserving the salient behavioural properties. Since logical models of regulatory networks can be represented as Petri nets [23], Petri net based methods could further be used to complement the reduction method described here. Complementary approaches in terms of formal abstractions have been proposed, including discrete abstractions of hybrid automata [24], or even the derivation of abstraction relationships between stochastic, discrete and Boolean semantics [25].

One particular feature of the reduction method defined here is that the removal of (functional) auto-regulatory components is forbidden. This rule is related to previous work on the dynamical roles of the regulatory circuits. Indeed, it has been recently proven in the discrete framework that positive regulatory circuits are necessary to generate multiple attractors, whereas negative circuits are necessary to generate cyclic attractors (cf. [26] and references therein). At least in the discrete framework, these properties depend only on the sign of the regulatory circuit, *i.e.* on the product of the signs of the involved interactions. From a qualitative dynamical point of view, it is thus possible to reduce the number of components of a circuit down to a single autoregulated component, while keeping the corresponding property, as long as we conserve the sign of the circuit (along with some functionality constraints).

Our formal presentation of the reduction method mainly focuses on the removal of a single component. However, this process can be iterated to remove several components. This raises the question of the impact of the order in which reductions are performed. Indeed, it can happen that a reduction of several components can be performed in one order and not in another one. This is because, in the course of multiple reductions, a regulatory circuit might be shortened up to a functional autoregulation (cf. Section 2.3), thus preventing further reductions. In contrast, the removal of a component might suppress an autoregulation on one of its targets, which can then be candidate to reduction. Considering the reduction of a set of components, when several orders are feasible, the obtained reduced logical regulatory graphs are equal and the dynamics

should thus be identical. If we aim at removing as many components as possible, the order of the removals of the components may be crucial. Further work is needed to properly define optimal or maximal reductions for the general case.

The worst case complexity of the algorithm for the reduction of a node $r$ that regulates $k$ targets is in $O(m^d)$, where $m$ is the highest number of levels of the involved components and $d$ is the depth of the MDDs representing the revised logical functions associated with the target nodes. In most cases, $m \leq 3$ and $d \leq 5$.

Applying our reduction method to a detailed model of the segment polarity network, we were able to show the absence of non-trivial attractors in a state transition graph too large to be efficiently analysed (the complete two cells model generates roughly $45 * 10^{12}$ states, while the reduced model state space has $15 * 10^6$ states). However, even if the reduction method dramatically reduces the size of the state transition graphs, it could be combined with other methods. For example, rather than considering monolithic models, a promising strategy would rely on the intrinsic modularity of these interconnected intracellular networks, developing compositional analyses.

The proposed reduction method offers a great flexibility to the modeller. Biological arguments (*e.g.* information on relative reaction speeds) can be used to select sets of nodes for consistent model reduction. In the course of the dynamical analysis of complex networks (*e.g.* multicellular networks), alternative reductions can be performed for specific aims, *e.g.* to ease the identification of all attractors or check their reachability from specific initial states.

To ease the maintenance of a detailed model along with its reduced versions, a novel *GINsim* release allows the user to define and record various reductions for the same reference model. This reduction method could further be combined with algorithmic methods enabling the compression and analysis of large state transition graphs ([27] and references therein), or even with model checking techniques ([28] and references therein), thereby facilitating the analysis of yet larger and more complex networks.

*Supplementary materials*

*GINsim* can be downloaded from http://gin.univ-mrs.fr/GINsim. The models are available in the *model repository* section of the same website.

## Acknowledgements

A.N. has been supported by a Ph.D. 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 (project ANR-08-SYSC-003), from EU FP7 (APO-SYS Project), and by the Belgian Science Policy Office (IAP BioMaGNet).

## References

[1] J. Saez-Rodriguez, L. Simeoni, J. Lindquist, R. Hemenway, U. Bommhardt, B. Arndt, U. Haus, R. Weismantel, E. Gilles, S. Klamt, B. Schraven, A logical model provides insights into *t* cell receptor signaling, PLoS Comput. Biol. 3 (2007) e163.
[2] R. Franke, M. Müller, N. Wundrack, E.-D. Gilles, S. Klamt, T. Kähne, M. Naumann, Host-pathogen systems biology: logical modelling of hepatocyte growth factor and helicobacter pylori induced c-met signal transduction, BMC Syst. Biol. 2 (2008) 4.
[3] C. Chaouiya, E. Remy, B. Mossé, D. Thieffry, Qualitative analysis of regulatory graphs: a computational tool based on a discrete formal framework, Lect. Notes Comput. Inf. Sci. 294 (2003) 119–126.
[4] A. Naldi, D. Berenguier, A. Fauré, F. Lopez, D. Thieffry, C. Chaouiya, Logical modelling of regulatory networks with GINsim 2.3, BioSystems 97 (2009) 134–139.
[5] R. Thomas, Regulatory networks seen as asynchronous automata: a logical description, J. Theoret. Biol. 153 (1991) 1–23.
[6] R. Thomas, D. Thieffry, 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 (1995) 247–276.
[7] A. Naldi, D. Thieffry, C. Chaouiya, Decision diagrams for the representation and analysis of logical models of genetic networks, Lect. Notes Comput. Sci. 4695 (2007) 233–247.
[8] R.E. Bryant, Graph-based algorithms for boolean function manipulation, IEEE Trans. Comput. 35 (1986) 677–691.
[9] T. Kam, T. Villa, R.K. Brayton, A.L. Sangiovanni-Vincentelli, Multi-valued decision diagrams: theory and applications, Int. J. Multiple-Valued Logic 4 (1998) 9–12.
[10] H. Meinhardt, Hierarchical inductions of cell states: a model for segmentation in drosophila, J. Cell Sci. 4 (Suppl.) (1986) 357–381.
[11] G. von Dassow, E. Meir, E.M. Munro, G.M. Odell, The segment polarity network is a robust developmental module, Nature 406 (2000) 188–192.
[12] N.T. Ingolia, Topology and robustness in the drosophila segment polarity network, PLoS Biol. 2 (2004) e123.
[13] R. Albert, H.G. Othmer, The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster, J. Theoret. Biol. 223 (2003) 1–18.
[14] M. Chaves, R. Albert, E.D. Sontag, Robustness and fragility of boolean models for genetic regulatory networks, J. Theoret. Biol. 235 (2005) 431–449.
[15] L. Sanchez, C. Chaouiya, D. Thieffry, Segmenting the fly embryo: logical analysis of the role of the segment polarity cross-regulatory module, Int. J. Dev. Biol. 52 (2008) 1059–1075.
[16] A. González, C. Chaouiya, D. Thieffry, Logical modelling of the role of the hh pathway in the patterning of the drosophila wing disc, Bioinformatics 24 (2008) i234–240.
[17] L. Sánchez, D. Thieffry, Segmenting the fly embryo: a logical analysis of the pair-rule cross-regulatory module, J. Theoret. Biol. 224 (2003) 517–537.
[18] A. Fauré, A. Naldi, C. Chaouiya, D. Thieffry, Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle, Bioinformatics 22 (2006) e124–131.
[19] J. Sifakis, Property preserving homomorphisms of transition systems, Lect. Notes Comput. Sci. 164 (1984) 458–473.
[20] G. Berthelot, Lri-lie, Checking properties of nets using transformation, Lect. Notes Comput. Sci. 222 (1986) 19–40.

[21] S. Haddad, J.-F. Pradat-Peyre, New efficient Petri nets reductions for parallel programs verification, Parallel Process. Lett. 16 (2006) 101–116.
[22] R. Sloan, U. Buy, Reduction rules for time Petri nets, Acta Inform. 33 (1986) 687–706.
[23] C. Chaouiya, A. Naldi, Remy, D. Thieffry, Petri net representation of multi-valued logical regulatory graphs, Nat. Comput. (available online), doi:10.1007/s11047-010-9178-0.
[24] R. Ghosh, A. Tiwari, C. Tomlin, Automated symbolic reachability analysis; with application to delta-notch signaling automata, Lect. Notes Comput. Sci. 2623 (2003) 233–248.
[25] F. Fages, S. Soliman, Abstract interpretation and types for systems biology, Theoret. Comput. Sci. 403 (2008) 52–70.
[26] E. Remy, P. Ruet, From minimal signed circuits to the dynamics of boolean regulatory networks, Bioinformatics 24 (2008) i220–226.
[27] A. Garg, A. Di Cara, I. Xenarios, L. Mendoza, G. De Micheli, Synchronous versus asynchronous modeling of gene regulatory networks, Bioinformatics 24 (2008) 1917–1925.
[28] P.T. Monteiro, D. Ropers, R. Mateescu, A.T. Freitas, H. de Jong, Temporal logic patterns for querying dynamic models of cellular interaction networks, Bioinformatics 24 (2008) i227–233.