



# Reactive Models for Biological Regulatory Networks

Daniel Figueiredo<sup>1</sup>(✉) and Luís Soares Barbosa<sup>2</sup>

<sup>1</sup> CIDMA, University of Aveiro, Aveiro, Portugal  
[daniel.figueiredo@ua.pt](mailto:daniel.figueiredo@ua.pt)

<sup>2</sup> HASLab INESC TEC and QuantaLab, University of Minho, Braga, Portugal

**Abstract.** A reactive model, as studied by D. Gabbay and his collaborators, can be regarded as a graph whose set of edges may be altered whenever one of them is crossed. In this paper we show how reactive models can describe biological regulatory networks and compare them to Boolean networks and piecewise-linear models, which are some of the most common kinds of models used nowadays. In particular, we show that, with respect to the identification of steady states, reactive Boolean networks lie between piecewise linear models and the usual, plain Boolean networks. We also show this ability is preserved by a suitable notion of bisimulation, and, therefore, by network minimisation.

**Keywords:** Biological regulatory networks · Switch graphs · Reactivity

## 1 Introduction

Biochemical processes occurring within cells are abstracted into the concept of biological regulatory networks. In general, such networks capture the cell dynamics, expressed as the concentration of each component (typically proteins, and other nucleotides), which are directed by the biochemical reactions occurring between them. This process is generally regulated by the DNA through the transcription of mRNA.

*Example 1.* An example of a “cascade” [6] is depicted in Fig. 1. This illustrates a simple and partial biological regulatory network organised as a sequence where each node induces the production of the following one.

One of the main goals of studying biological regulatory networks is the identification of *steady states* which somehow represent the “way of working” of a cell. Different steady states can be associated, for example, to the differentiated cells of a living organism sharing the same DNA.

Models for biological regulatory networks fall in one of two major classes [7]: quantitative – where the exact concentration of each component in a cell is given; or *qualitative* – focussed on the overall dynamics of a system, and classifying the



**Fig. 1.** A example of a “cascade”.

concentration of a component in qualitative terms, such as “high” or “low”. Quantitative models are, of course, more precise, but also harder to manipulate. A “hybrid” approach to this problem, in which a qualitative model is used to explain the “big picture” and quantitative models are applied in a second stage to study the detailed dynamics of a behavioural region [5] is a typical compromise.

This paper introduces a new qualitative model into the picture—*reactive Boolean networks* which seems particularly interesting to help in the quest for steady states, a major issue in the area. The model builds on the notion of *reactivity* [11] which proved successful in both fundamental research in modal logic [10, 12, 14] and software engineering applications [1, 3, 9].

**Outline.** Section 2 provides the background for this piece of research by briefly revisiting the most common models for biological regulatory networks. The notion of reactivity and the associated formal structures, namely, switch graphs and reactive frames are discussed in Sect. 3. Section 4 contains the paper main contribution, introducing reactive Boolean networks and stating some of the relevant properties. Finally, Sect. 5 concludes and identifies some directions for future work.

## 2 Biological Regulatory Networks

Several modelling approaches have been proposed to formally characterise biological regulatory networks [7]. This background section revisits some of them to set the scene for our own proposal discussed in Sect. 4. In general, models for biological regulatory networks consider components  $i = 1, \dots, n$ , representing e.g. protein, genes, mRNA, and variables  $x_1, \dots, x_n$  corresponding to the concentration or level of expression of the respective component.

**Ordinary Differential Equations and Piecewise Linear Models.** The classical quantitative models resort to ordinary differential equations. The interaction between components is captured by sigmoid expressions embedded in differential equations. Either positive or negative regulations of a component  $i$  over a component  $j$  (meaning that component  $i$  respectively induces or inhibits the production/activation of component  $j$ ) may be considered. This is achieved through a quantitative representation of the system. One can consider variables  $x_i$  which are related to each component  $i$ . A variable  $x_i$  represent the concentration of the component  $i$  in the system and we can use them to represent the (either positive or negative) regulations occurring between these components. This is accomplished *via* the introduction of a sigmoid function depending on each  $x_j$  in function  $f$ , in the context of the differential equation  $x_i = f(x_1, \dots, x_n)$  describing the concentration of component  $i$ .

Several classes of sigmoid functions can be chosen. A common alternative takes the form of a fraction  $s^+(x; \theta, n) = \frac{x^n}{\theta^n + x^n}$  for positive regulations and  $s^-(x; \theta, n) = 1 - s^+(x; \theta, n) = \frac{\theta^n}{\theta^n + x^n}$  for negative ones.

In this context, a model is obtained by considering a system with ODEs with the form  $x_i = F_i(x_1, \dots, x_n) - \gamma_i x_i$ , where each  $F_i(x_1, \dots, x_n)$  is obtained as sums and products of the referred sigmoid functions and reflects how the interaction between components affect the production/expression of  $i$ . Above  $\theta$  is a threshold determining the concentration of  $x$  needed to effectively regulate the target component and  $n$  determines how abrupt this regulation changes from almost inexistent to effective. Also,  $\gamma_i$  is a constant representing the degradation rate of the component  $i$ . All thresholds  $\theta$ ,  $n$  in the sigmoid functions and  $\gamma$  are usually estimated using suitable methods.

As one would expect, this sort of models containing non linear differential equations are hard to study analytically, but used in a number of contexts to simulate and predict the answer of a biological system. To overcome the challenge of solving a system of non linear ODEs, a more manageable alternative divides the entire state space into a finite number of domains and studies each one in its own. On a second stage the different domains are integrated and the general dynamics of the system recovered. Such models are called *piecewise linear* (PWL).

In practice, to obtain a piecewise linear model from a system of differential equations specifying a biological regulatory network, one ignores the estimated value for  $n$  and assumes that  $n \rightarrow +\infty$ , *i.e.*

$$\frac{x^n}{\theta^n + x^n} \xrightarrow{n \rightarrow +\infty} \begin{cases} 1, & \text{if } x > \theta \\ \frac{1}{2}, & \text{if } x = \theta \\ 0, & \text{if } x < \theta \end{cases}$$

Thus, the state space is divided into two different domains ( $x < \theta$  and  $x > \theta$ ) and a boundary  $x = \theta$ . Given a specific model, this technique is applied to each one of these sigmoid functions in order to split the state space into several regions with a linear differential equation expressing the state trajectory within each one.

*Example 2.* Consider the following system of differential equations:

$$\begin{cases} x' = 5 \frac{x^2}{x^2 + 2^2} \cdot \frac{2^2}{y^2 + 2^2} - x \\ y' = 3 \frac{x^2}{x^2 + 4^2} - y \end{cases}$$

Making  $n \rightarrow +\infty$  leads to

$\begin{cases} x' = -x \\ y' = -y \\ x < 2 \\ 2 < y \end{cases}$	$\begin{cases} x' = -x \\ y' = -y \\ 2 < x < 4 \\ 2 < y \end{cases}$	$\begin{cases} x' = -x \\ y' = 3 - y \\ 4 < x \\ 2 < y \end{cases}$
$\begin{cases} x' = -x \\ y' = -y \\ x < 2 \\ y < 2 \end{cases}$	$\begin{cases} x' = 5 - x \\ y' = -y \\ 2 < x < 4 \\ y < 2 \end{cases}$	$\begin{cases} x' = 5 - x \\ y' = 3 - y \\ 4 < x \\ y < 2 \end{cases}$

Analytically, a steady state can be identified at point  $(0, 0)$ , and an orbit which asymptotically converges to  $(4, 2)$ .

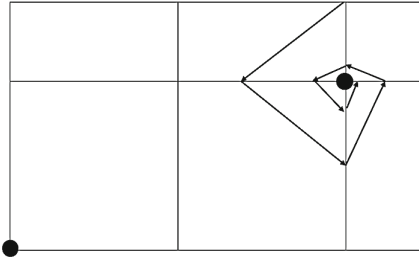


Fig. 2. Steady states identified in Example 2.

Formally, it is important to mention that, in general, a more careful study at the domains should be performed. However, we do not focus much on this issue for now because we will only consider PWL models where no such care is required.

In the context of a differential equation, we say that a *flow* is a trajectory obtained from a initial state which is ruled by the differential equations. In the context of a PWL model, we give a similar meaning to this term but we admit a flow to be the concatenation of several usual flows, obtained within different domains, regarding that this concatenated flow still is continuous. For instance, in Fig. 2, a flow in a PWL model is illustrated by a solid line. The orientation represented by the arrows describes the evolution of the flow along time.

**Boolean Networks.** A Boolean network (BN) is another kind of model to describe the dynamics of biological regulatory networks. This kind of model considers the concentrations  $x_i$  as Boolean variables assuming, in practice,  $x_i = 0$  if the concentration of  $i$  is “low” (bellow some threshold) and  $x_i = 1$  if the concentration of  $i$  is “high” (above some threshold).

In a BN model, each variable  $x_i$  is regulated by a Boolean function  $f_i(x_1, \dots, x_n)$  built from Boolean operators, combined in a general function

$f = (f_1, \dots, f_n)$ . From this function, a graph (Boolean network)  $(V, E)$  can be obtained. At this point, two paradigms can be considered: synchronous and asynchronous. For the synchronous approach, the BN is obtained as follows: given the set of components  $A$  with  $|A| = n$ ,  $V = \{0, 1\}^n$  and a directed edge  $(a, b) \in E$  if  $b = f(a)$ . In this way, each vertex in  $V$  admits exactly one output. For the asynchronous approach  $V$  is obtained as before but a directed edge  $(a, b) \in E$  if there is some index  $i$  such that  $b_i = f(a)_i \neq a_i$  and  $b_j = a_j$  for every other indexes  $j \neq i$ . In this way, only the value of a single variable is updated at each step. Thus, we admit networks where several edges can have the same vertex as tail and where vertices with no outgoing edges are allowed. The asynchronous approach was proposed and is often used since it reflects the dynamics of the system in a more realistic way.

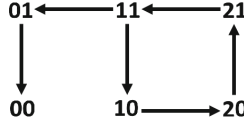
For our approach we do not need to consider these theoretical concepts regarding BNs in detail, since our network will be built using a different procedure. Thus, it is enough to consider the underlying graphs as BN models, although a wider connection with the remaining theory of BNs could be established. For more information about this, we recommend the reader to consult [13] for more details.

A qualitative perspective, as captured by a Boolean network, can be obtained from a piecewise linear model. For this, a graph  $(V, E)$  is built taking the domains of the piecewise linear model as the states  $V$ , and identifying the edges with the flows, *i.e.* an edge from  $i$  to  $j$  is added to  $E$  if there is a flow from the domain  $i$  to the domain  $j$  in the piecewise model. In this paper we consider asynchronous dynamics, meaning, in practice, that we only admit “adjacent” domains to be connected.

As referred, in BNs, variables can take values 1 or 0 depending on the concentration/level of expression of the corresponding protein/gene being, respectively, high or low (above or below a threshold  $\theta$ ). Since in PWL models we consider thresholds to split the state space, these Boolean variables are then used to identify each domain. However, in order to be able to fully accomplish this, sometimes more than two values must be considered. For instance, a third value may be added to cater for intermediate states. This will be the case in the next example which illustrates how we need to consider three possible values for  $x$ —0, 1 and 2—for low, medium and high concentration, respectively.

*Example 3.* Figure 3 shows a BN corresponding to the biological regulatory network described in Example 2. We can think of it as a graph where the vertices have the form  $ab$  where  $a$  and  $b$  represent the level of the corresponding component. We note that we must consider three values for the variable related to  $a$  since in the PWL model two thresholds are considered for the variable  $x$ . For instance state 00 represents the state where both components have low concentration and 10 is the one whose first component has a medium concentration and the second component has a low concentration. Recalling its piecewise linear model, we note that each vertex corresponds to a domain and the edges are introduced according to the flows within each domain. For instance, there is an edge from 11 to 01 because, in the corresponding PWL model it is possible to

attain the boundary between the domain represented by 01 and 11 with a flow whose initial state is  $(x, y) = (\frac{5}{2}, \frac{7}{2})$ , which is within the domain denoted by 11.



**Fig. 3.** Example of a Boolean network.

Steady states in a BN are identified by strongly connected components with no outgoing edges [7], which is called an *attractor* in the literature. Given a graph  $(V, E)$ , a strongly connected component is a set of vertices  $X$  such that, for every  $x, y \in X$ , there exists a directed path between  $x$  and  $y$  which only contains vertices in  $X$ .

The BN depicted in Example 3 admits two strongly connected components –  $\{00\}$  and  $\{11, 21, 20, 10\}$  – but only one of them is an attractor and, thus, represents a steady state, since  $\{11, 21, 20, 10\}$  admits an outgoing edge from 11 to 01. Actually, it is usual to lose information when going from a piecewise linear model to a Boolean network. In general, it is well known that, for a system  $S$ ,

$$\text{SteadySt}(\llbracket S \rrbracket_{BN}) \subseteq \text{SteadySt}(\llbracket S \rrbracket_{PWL}) \subseteq \text{SteadySt}(\llbracket S \rrbracket_{ODE})$$

where  $\llbracket S \rrbracket_M$  refers to the representation of system  $S$  in model  $M$  and *SteadySt* retrieves the set of steady states in a model.

As expected, simpler models tend to lose information about the steady states of the system. Our proposal, discussed in the sequel tries to partially correct this problem.

### 3 Switch Graphs and Reactive Frames

As mentioned in Sect. 1, the model proposed in this paper borrows from modal logic research the notion of *reactive frame*. Modal logics [4] are logics in which formulae are interpreted over a graph of semantic universes interconnected through an accessibility relation. The former may represent *e.g.* temporal instants, deontic contexts or epistemic states. In all cases truth is relative rather than universal as in classical logics which assume just one universe of interpretation. Modal operators— $\Box$  and  $\Diamond$ —provide a universal, respectively, existential, quantification over the accessible universes from the current point of evaluation.

Typically this accessibility relation (*i.e.* the underlying graph) is fixed. A *reactive frame*, however, is a graph whose structure can vary over time. Such graphs have been used as semantic models for some classes of modal logics in which the way semantic universes are interconnected can be modified on the fly.

The idea can be traced back to Johan van Benthem seminal paper on the so-called sabotage logic [3] in which an edge is deleted after being taken (therefore preventing its subsequent use). In another variant described by Areces *et al.* [1], edges are not deleted but their direction swapped. This sort of logics [2] and the underlying dynamic graphs are called *reactive* as they capture structural changes under reaction to previous behaviour.

Reactive behaviour is often found in games. For instance, castling, a chess special move involving the King and a Rook, can only be performed if none of the pieces which take part has been moved before. Therefore, in identical configurations of the table, different moves can be possible. A similar situation involves another chess move called *en passant*, which allows a player to capture a pawn from the opponent with one of his pawns whenever the opponent moves his pawn two squares in front and it becomes laterally adjacent to an enemy pawn. However, if the opponent only moves his pawn a square in front, *en passant* becomes illegal.

### 3.1 Switch Graphs

Syntactically, such moving structures are represented by *switch graphs* [10,12], which add higher-level edges to the usual graph structure. These higher-level edges connect basic edges, also called “0-level” edges, which are the ones which are eventually crossed, to other higher-level edges according to the following definition.

**Definition 1.** *Given a set of nodes  $W$ , a switch graph is a pair  $(W, S)$  where  $S = \bigcup_{n \geq 0} S_n$  such that:*

- $S_0 \subseteq W \times W$ , i.e. *the usual relation between nodes,*
- *and, for  $n \geq 1$ ,  $S_n \subseteq S_0 \times S_{n-1} \times \{\circ, \bullet\}$ .*

A higher-level edge  $(d, e, *)$  will either inhibit or activate its target edge  $e$  whenever the source edge  $d$  is crossed, depending on the value of annotation  $*$ . Target edge  $e$  will be inhibited if  $* = \circ$  or activated if  $* = \bullet$ . In the graphical representation of a switch graph, as shown in Fig. 4, inhibitor edges are depicted as white headed arrows, while black headed arrows represent activator edges.

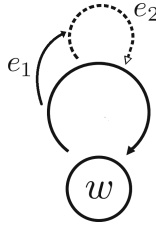
A switch graph is configured through an *instantiation* function  $I : S \rightarrow \{0, 1\}$  which marks each edge as inhibited or active depending on  $I(s) = 0$  or  $I(s) = 1$ , respectively. The former (respectively, latter) edges are depicted as dashed (respectively, full) arrows. Note that inhibited edges cannot be crossed, and they can neither activate nor inhibit other edges. Moreover, only 0-level edges can be crossed: if one such edge  $x$  is crossed, all active higher-level edges with source in  $x$ , i.e.  $(x, e, *)$  will fire and activate/inhibit the respective target edge  $e$ .

*Example 4.* Figure 4 depicts a switch graph  $(W, S)$  with  $W = \{w\}$  and

$$S = \{(w, w), ((w, w), (w, w), \circ), ((w, w), ((w, w), (w, w), \circ)), \bullet\}$$

For simplicity, we define  $e_1 = ((w, w), ((w, w), \circ)), \bullet$  and  $e_2 = ((w, w), (w, w), \circ)$ . The initial instantiation  $I_0$  is such that  $I_0(w, w) = 1$  (the edge  $(w, w)$  can be crossed),  $I_0(e_2) = 0$  (meaning that it is inhibited) and  $I_0(e_1) = 1$  (therefore, activated and ready to activate  $e_2$ , the pointed edge whenever  $(w, w)$  is crossed).

Therefore, starting from  $w$ , the edge  $(w, w)$  can be crossed (since it is active) and this causes the higher-level edge  $e_1$  to fire and activate  $e_2$ .  $e_2$  has no effect since it was initially inhibited when  $(w, w)$  was crossed. One can then cross  $(w, w)$  again. Now,  $e_1$  acts but has no effect, since  $e_2$  is already active, while  $e_2$  acts and inhibits  $(w, w)$ . Hence,  $(w, w)$  can no longer be crossed. This switch graph shows an example of counter which can “count” only twice.



**Fig. 4.** Example of a switch graph.

### 3.2 From Switch Graphs to Reactive Frames

A switch graph can be encoded by the set of paths it generates, as shown in the following example.

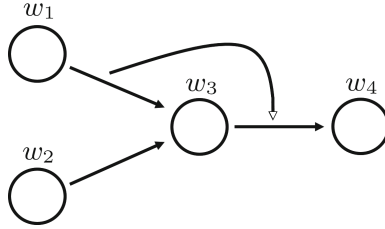
*Example 5.* The set of paths  $\Delta$  corresponding to the switch graph shown in Fig. 5 is generated as follows. Starting at  $w_1$  we can move to  $w_3$  but not from there since the edge  $(w_3, w_4)$  was deleted on crossing  $(w_1, w_3)$ . Thus,  $(w_1), (w_1, w_3) \in \Delta$ . Starting at  $w_2$  we can move to  $w_3$  and afterwards to  $w_4$  (since the edge  $(w_3, w_4)$  was preserved). Thus,  $(w_2), (w_2, w_3), (w_2, w_3, w_4) \in \Delta$ . Starting at  $w_3$  there is a move to  $w_4$ , from where no other move is possible. Therefore,  $\Delta = \{(w_1), (w_1, w_3), (w_2), (w_2, w_3), (w_2, w_3, w_4), (w_3), (w_3, w_4), (w_4)\}$ .

Paths are used to define *reactive frames*—a semantic model for switch graphs upon which suitable (reactive) modal logics are defined. We will not develop such logics in this paper, the interested reader is referred to [10,12] for an extensive account.

**Definition 2.** Consider  $W \neq \emptyset$  and let  $\Delta \subseteq W^*$  be a nonempty set of finite paths.  $(W, \Delta)$  is a reactive frame if:

- $(w) \in \Delta$  for any  $w \in W$
- $\forall n \geq 1, (w_1, \dots, w_n, w_{n+1}) \in \Delta$  implies  $(w_1, \dots, w_n) \in \Delta$ .





**Fig. 5.** Another switch graph.

Let us end this section, fixing some notation. As usual, we denote by  $W^*$  the set of all non-empty finite sequences (*paths*) over set  $W$ , and define function  $t : W^* \rightarrow W$  by  $t(\lambda) = t(w_1, \dots, w_n) = w_n$ . Let  $\lambda = (w_1, \dots, w_n)$  be a path. Notation  $\lambda w$  abbreviates the path  $(w_1, \dots, w_n, w)$ . Similarly, if no ambiguity arises,  $w_1$  stands for the path  $(w_1)$ . Finally, a path  $\gamma$  *extends* or *is an extension* of a path  $\lambda$  if there exist  $w_0, \dots, w_n \in W$  such that  $\gamma = \lambda w_0 \dots w_n$ . Of course, every path is an extension of itself.

## 4 Reactive Boolean Networks

Switch graphs provide an interesting alternative to represent biological regulatory networks, building on the corresponding Boolean networks. As discussed below, this new model which we propose to call *reactive Boolean networks* (RBN) has a number of interesting properties namely in what concerns the identification of steady states and their preservation under model minimisation.

In the next definition, we consider a PWL model and the corresponding Boolean network obtained from it. There, when we mention a vertex  $x$  of the Boolean network we also mention the corresponding domain of the PWL model and *vice-versa*.

**Definition 3.** *Given a PWL model  $M$  whose corresponding Boolean network is  $N$ , a reactive boolean network is a two-level switch graph  $(W, S)$  where  $(W, S_0) = N$  and  $S_1$  is obtained according to the following rules:*

1. *For any domain  $k$  of  $M$  such that  $u = (j, k) \in S_0$ , we have  $(v, u, \circ) \in S_1$  with  $v = (i, j)$  if a flow which enters in region  $j$  via the boundary between regions  $i$  and  $j$  never leaves it via the boundary between regions  $j$  and  $k$ .*
2. *For each  $(v, u, \circ) \in S_1$  with  $u = (j, k)$  and  $v = (i, j)$ , then  $(w, u, \bullet) \in S_1$  if there exists  $w = (l, j) \in S_0$  for some region  $l$  of  $M$  such that there is a flow entering in region  $j$  via the boundary between regions  $l$  and  $j$  and leaving it via the boundary between  $j$  and  $k$ .*

*Moreover, since  $(W, S)$  is a two-level switch graph, then  $S_n = \emptyset$ , for  $n > 1$ .*

We only define Reactive Boolean networks for two-level switch graphs. In fact, this definitions could be generalized to embed higher-level edges but we

believe that the benefits would not be worth the additional computational cost. Then, we leave this generalization for future work.

In practice, this new kind of models can temporarily delete from the state transition graph edges that would represent non-realistic behaviours. In practice, and since we can compute the flow given by a linear differential equation and an initial state, the inclusion of an edge of the type  $((i, j), (j, k), \circ) \in S_1$  means that is not possible to obtain, in the PWL model, a flow with initial state in the region  $i$  that enters in the region  $j$  and leads us to region  $k$ .

#### 4.1 Recovering Attractors

As mentioned before, it is well known that, in general:

$$\text{SteadySt}(\llbracket S \rrbracket_{BN}) \subseteq \text{SteadySt}(\llbracket S \rrbracket_{PWL})$$

with equality failing for multiple examples. Reactive Boolean networks, on the other hand, in general can increase the number of steady states that can be identified when comparing to Boolean networks, introducing a further level in this inequality:

$$\text{SteadySt}(\llbracket S \rrbracket_{BN}) \subseteq \text{SteadySt}(\llbracket S \rrbracket_{RBN}) \subseteq \text{SteadySt}(\llbracket S \rrbracket_{PWL})$$

In the context of reactive Boolean networks, steady states are also identified by *attractors*, whose definition is revised as follows.

**Definition 4.** *Given a reactive Boolean network  $(W, S)$  whose set of paths is  $\Delta$ , a set  $V \subseteq W$  forms a strongly connected component relatively to a path  $\lambda \in \Delta$  ( $SCC_\lambda$ ) if for any  $v \in V$  and any path  $\rho \in \Delta$  which extends  $\lambda$ , there exists  $\gamma \in \Delta$  such that  $t(\gamma) = v$  and  $\gamma$  extends  $\rho$ .*

**Proposition 1.** *If  $V$  is a  $SCC_\lambda$ , it is always possible to find a path between two states  $u, v \in V$  after a reconfiguration on the edges, induced by the path  $\lambda$ .*

*Proof.* From the definition, for all extensions  $\rho$  of  $\lambda$ , one can find  $\lambda$  an extension of  $\rho$  such that  $t(\gamma_u) = u$ . Again, by definition, and since  $\gamma_u$  is itself extends  $\lambda$ , it is possible to find  $\gamma_v$  which extends  $\gamma_u$  and such that  $t(\gamma_v) = v$ .  $\square$

**Definition 5.** *Given a reactive Boolean network  $(W, S)$  whose set of paths is  $\Delta$ , a set  $V \subseteq W$  is an attractor if it is a  $SCC_\lambda$ , for some path  $\lambda$ , and every path  $\gamma$  extending  $\lambda$  verifies  $t(\gamma) \in V$ .*

This definition extends the notions of SCC and attractor, which are defined for regular graphs (and BNs) to switch graphs (and RBNs). Informally, this means that a set  $V$  is an attractor if there is a path  $\gamma$  such that, after walking along it, we can always find a path between any two states of  $V$  and there is not any path guiding us to a final state outside the set  $V$  (*i.e.*, the usual definition for attractor in a usual graph).

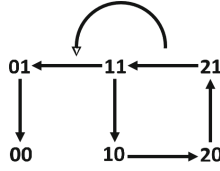


Fig. 6. A reactive Boolean network.

Example 6. Recall Examples 2 and 3, and take the Boolean network introduced then as the first stage  $(W, S_0)$  of a reactive Boolean network. The piecewise linear model in Example 2 generates  $S_1$  which turns out to be a singleton relation  $\{((21, 11), (11, 01), \circ)\}$ . The whole net is depicted in Fig. 6.

For this reactive network, given 21 as the initial state, we obtain the following set of paths:  $\{(21), (21, 11), (21, 11, 10), (21, 11, 10, 20), (21, 11, 10, 20, 21), \dots\}$ . Therefore, according to Definition 5,  $\{11, 10, 20, 21\}$  is an attractor for this reactive Boolean network. Similarly, taking 00 as the initial state,  $\{00\}$  emerges as an attractor as well.

**Definition 6.** Consider a BN or RBN model  $(V, E)$  and a RBN model  $(V', E')$ . Given an attractor  $A \subseteq V$  of the model  $(V, E)$ , we say that it is signaled by the model  $(V', E')$  if there is an algorithm that allows us to obtain a set  $B \subseteq V'$  from  $A$  in an unambiguous way and such that  $B$  is an attractor of  $(V', E')$ .

Using other words, if a model “A” signals the attractors of another model “B”, we can recover all attractors of the model “B” from the attractors of the model “A”.

**Proposition 2.** Given a piecewise linear model, the corresponding reactive Boolean network identifies, in general, a larger set of attractors than the simpler Boolean network. Moreover, all attractors of a BN are signalled in the corresponding RBN.

*Proof.* The fact that, in general, a RBN a larger set of attractors from than a BN was already shown in Example 6. Note that we were able to recover the attractor 00 as well as the converging cyclic behaviour of the piecewise linear model that the BN was not able to signal.

Now, consider a BN with an attractor  $V$ . If  $|V| = 1$ , then  $V$  is also attractor of the corresponding RBN. Otherwise, let  $\lambda = (v)$  be a path with  $v \in V$ . Since  $V$  is an attractor in a BN, all extensions of  $\lambda$  terminate at elements of  $V$ . Consider the following algorithmic procedure: Choose  $v \in V$  for which it is possible to consider an extension  $\gamma$  of  $\lambda$  such that it is no more possible to extend it to path  $\rho$  where  $t(\rho) = v$ . If such a path exists, update  $\lambda$  to  $\gamma$  and  $V$  to  $V \setminus \{v\}$ . Repeat this process while it is possible to choose such a  $v$ . Note that, since  $V$  and  $S$  are finite (*i.e.* there is a finite number of configurations for the  $S_0$  edges), this algorithm terminates. Note that, after this process, for any  $v \in V$  such that there is an extension  $\gamma_v$  of  $\lambda$  such that  $t(\gamma_v) = v$ , and for every  $w \in V$ , there

is an extension  $\rho_w$  of  $\gamma_v$  such that  $t(\rho_w) = w$ . This proves that  $V$  is a  $\text{SCC}_\lambda$  and, since each extension  $\gamma$  of  $\lambda$  is such that  $t(\gamma) \in V$ , the RBN signals an attractor.  $\square$

**Proposition 3.** *All attractors in a RBN are steady states of the corresponding piecewise linear model.*

*Proof.* A steady state of a PWL can be found as either an invariant region or cyclic behaviour which asymptotically converges to a point or orbit. Let  $V$  be an attractor in a RBN, and consider a region  $T$  resulting from the union of every domain represented by  $i \in V$ . Since  $V$  is a  $\text{SCC}_\lambda$  for some path  $\lambda$ , this means that there is a flow in the piecewise linear model which makes impossible to leave region  $T$ . This means that there exists an invariant subregion  $T'$  of  $T$  and, therefore, it contains a steady state.  $\square$

## 4.2 Bisimulation and Minimisation

In a previous publication [9], the authors defined a notion of bisimulation for reactive frames and proved a Hennessy-Milner like theorem stating the equivalence between bisimilarity (*i.e.* the existence of a bisimulation relating two nodes in a frame), and logical validity (*i.e.* the fact that both nodes satisfy exactly the same set of formulæ expressed in a suitable modal logic). From the modelling point of view adopted in this paper, bisimulation is a crucial tool to reduce the size of a reactive frame while keeping the behaviour it may induce, thus increasing the performance of any automatic analysis tool operating over reactive frames. Some examples of application of bisimulation to non reactive models can be found in [8, 15]. Bisimulations can be also used for other purposes in a biological context: for example, in [8], attractors of a Boolean models are highlighted using bisimulations.

**Definition 7.** *Let  $(W, \Delta)$  and  $(W', \Delta')$  be two reactive frames. A relation  $\mathcal{S} \subseteq \Delta \times \Delta'$  is a bisimulation if and only if  $\forall \lambda \in \Delta, \forall \lambda' \in \Delta'$  such that  $(\lambda, \lambda') \in \mathcal{S}$ :*

*R-zig:*  $\forall w \in W (\lambda w \in \Delta \Rightarrow \exists w' \in W', \lambda' w' \in \Delta' \text{ such that } (\lambda w, \lambda' w') \in \mathcal{S})$

*R-zag:*  $\forall w' \in W' (\lambda' w' \in \Delta' \Rightarrow \exists w \in W, \lambda w \in \Delta \text{ such that } (\lambda w, \lambda' w') \in \mathcal{S})$

*P-zig:*  $\forall \gamma \in \Delta (t(\lambda) = t(\gamma) \Rightarrow \exists \gamma' \in \Delta' (t(\lambda') = t(\gamma') \text{ and } (\gamma, \gamma') \in \mathcal{S}))$

*P-zag:*  $\forall \gamma' \in \Delta' (t(\lambda') = t(\gamma') \Rightarrow \exists \gamma \in \Delta (t(\lambda) = t(\gamma) \text{ and } (\gamma, \gamma') \in \mathcal{S}))$

*Example 7.* Figure 7 depicts two switch graphs which induce bisimilar reactive models. In fact, we can verify that the following relation is a bisimulation:

$$\{((w_1), (v_1)), ((w_1, w_2), (v_1, v_2)), ((w_2), (v_2)), ((w_2, w_3), (v_2, v_2)), ((w_4), (v_1)), ((w_4, w_3), (v_1, v_2)), ((w_3), (v_2)), ((w_3, w_2), (v_2, v_2))\}$$

Bisimulation can also easily be formulated for switch graphs, and therefore for reactive Boolean networks, as follows.

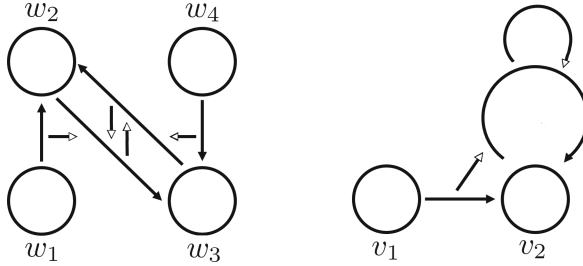


Fig. 7. Two switch graphs whose corresponding reactive frames are bisimilar.

**Definition 8.** Given two switch graphs  $(W, S)$ ,  $(W', S')$  whose reactive frames are  $(W, \Delta)$  and  $(W', \Delta')$ , and an equivalence relation  $\mathcal{R} \subseteq W \times W'$ , we say that a relation  $\mathcal{B} \subseteq \Delta \times \Delta'$  is induced by  $\mathcal{R}$  when  $\mathcal{B}$  is such that:

- $(w, w') \in \mathcal{R} \Leftrightarrow ((w), (w')) \in \mathcal{B}$  for every  $w \in W$ ,  $w' \in W'$ .
- Let  $\lambda \in \Delta$  and  $\lambda' \in \Delta'$  be such that  $(\lambda, \lambda') \in \mathcal{B}$ . For every  $w \in W$  and  $w' \in W'$  such that  $\lambda w \in \Delta$  and  $\lambda' w' \in \Delta'$ , we have  $\lambda w \mathcal{B} \lambda' w'$  iff  $(w, w') \in \mathcal{R}$ .

Moreover, we say that  $\mathcal{R}$  is a bisimulation iff the induced relation  $\mathcal{B}$  is a bisimulation for reactive models.

From a model  $(W, S)$ , we can soundly obtain a reduced model  $(W', S')$  if there exists a bisimulation  $R$  verifying  $\forall w \in W \exists w' \in W' w R w'$ . In this case, we say  $R$  is total. Our final result states that bisimulation preserve attractors.

**Lemma 1.** Let  $(W, S)$  and  $(W', S')$  be two switch graphs,  $(W, \Delta)$  and  $(W', \Delta')$  be corresponding reactive frames. Let  $\mathcal{R}$  be a total bisimulation between  $(W, S)$  and  $(W', S')$  and  $\mathcal{B}$  be the induced relation from  $\mathcal{R}$ . Then  $\mathcal{B}$  is total whenever  $\mathcal{R}$  is total.

*Proof.* We prove this lemma by induction over paths.

Let  $\lambda \in \Delta$  be a path. If  $\lambda = (w)$ , for some  $w \in W$ , then, since  $R$  is total, there is  $w' \in W'$  such that  $(w, w') \in \mathcal{R}$  and, therefore  $(w) \mathcal{B} (w')$ .

Let us now consider a path  $\gamma = \lambda w$  for some  $w \in W$  and  $\lambda \in \Delta$ . Then, by induction hypothesis, there are  $\lambda' \in \Delta'$  such that  $\lambda \mathcal{B} \lambda'$ . Then, since  $\mathcal{B}$  is a bisimulation and by definition,  $\exists w' \in W'$  such that  $\lambda w \mathcal{B} \lambda' w'$ . □

**Proposition 4.** Let  $(W, S)$  and  $(W', S')$  be two bisimilar switch graphs. Each attractor of  $(W, S)$  is signaled by some attractor of  $(W', S')$ .

*Proof.* Let  $(W, S)$  and  $(W', S')$  be two switch graphs whose corresponding reactive frames are  $(W, \Delta)$  and  $(W', \Delta')$ , respectively. Let also  $R \subseteq W \times W'$  be a total bisimulation and  $T$  the corresponding bisimulation for reactive frames.

Consider  $A$ , an attractor of  $(W, S)$ . Thus, there is some path  $\lambda \in \Delta$  such that  $A$  is a  $\text{SCC}_\lambda$ . According to the previous lemma and since  $\mathcal{R}$  is total, then  $\mathcal{B}$  is also total. Then, by definition, there is  $\lambda' \in \Delta'$  such that  $(\lambda, \lambda') \in T$ .

Let  $\bar{B} = \{t(\gamma') : \gamma' \in \Delta' \text{ be an extension of } \lambda'\}$ . By the definition of  $\bar{B}$ , and using a process analogous to the one presented in the proof of Proposition 2, we obtain an attractor  $B \subseteq \bar{B}$ . We will show that the states of  $B$  are related with the states of  $A$ .

If  $b \in B$ , then it means that there is an extension  $\gamma'$  of  $\lambda'$  such that  $(\gamma') = b$ , i.e.  $\exists w'_0, \dots, w'_n$  such that  $\gamma' = \lambda'w_0 \dots w_n$ . Since  $R$  is a bisimulation, we know that  $t(\lambda)Rt(\lambda')$ ,  $t(\lambda w_0)Rt(\lambda'w'_0)$ ,  $\dots$ ,  $t(\lambda w_0 \dots w_n)Rt(\lambda'w_0 \dots w'_n)$ , where  $w_0, \dots, w_n \in W$  are such that  $w_0Rw'_0, \dots, w_nRw'_n$ . Since  $A$  is an attractor  $w_0, \dots, w_n \in A$ .  $\square$

We end this paper with some considerations about the expressibility of switch graphs when compared with regular graphs. In fact, they present similar expressibility as when can think, in some sense, in a translation of switch graphs to usual graphs: a switch graph can be seen as a regular graph where each state is a pair  $(x, I)$  where  $x$  is a state of the switch graph and  $I$  is an admissible instantiation. The accessibility relation is defined such that there is an edge between two states  $(x, I)$  and  $(y, J)$  if  $I(x, y)$  is defined and it is equal to  $I$ , and  $I$  is updated to  $J$  when the edge  $(x, y)$  is crossed. Although this “translated” model is not so intuitive as the switch graph itself, it allows us to obtain a finite and usual graph from a switch graph  $(W, S)$  whenever  $S$  is finite. Therefore, it allows us to apply the already existing tools in Computer Science to study switch graphs.

## 5 Conclusions and Future Work

This paper proposed a new model for biological regulatory networks based on the notion of *reactivity* as introduced by Gabbay and his collaborators in the context of transition systems and their modal logics. The proposed model—*reactive Boolean networks*—is discrete and finite (therefore amenable to transformation to a plain graph representation). Similarly to the usual Boolean networks, the reactive ones provide a straightforward way to simplify piecewise linear models. However, as shown here, its ability to identify steady states overcomes usual Boolean networks. We also show that such an ability is preserved under a suitable notion of bisimulation and, therefore, under network minimisation.

There are several avenues for future work we are currently exploring. The first one consists of introducing weighted edges in the reactive network to capture either some form of uncertainty in the cell evolution or describe the consumption of contextual resources. We also intend to resort to the reactive modal logic [10] interpreted over reactive frames to formulate and verify properties of biological regulatory networks.

**Acknowledgments.** This work was supported by ERDF - The European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project POCI-01-0145-FEDER-030947. and project with reference UID/MAT/04106/2019 at

CIDMA. D. Figueiredo also acknowledges the support given by FCT via the PhD scholarship PD/BD/114186/2016.

The authors are also grateful to the reviewers for their useful comments and corrections.

## References

1. Areces, C., Fervari, R., Hoffmann, G.: Swap logic. *Log. J. IGPL* **22**(2), 309–332 (2013). p. jzt030
2. Areces, C., Fervari, R., Hoffmann, G.: Relation-changing modal operators. *Log. J. IGPL* **23**(4), 601–627 (2015). p. jzv020
3. van Benthem, J.: An essay on sabotage and obstruction. In: Hutter, D., Stephan, W. (eds.) *Mechanizing Mathematical Reasoning*. LNCS (LNAI), vol. 2605, pp. 268–276. Springer, Heidelberg (2005). [https://doi.org/10.1007/978-3-540-32254-2\\_16](https://doi.org/10.1007/978-3-540-32254-2_16)
4. Blackburn, P., De Rijke, M., Venema, Y.: *Modal Logic: Graph. Darst*, vol. 53. Cambridge University Press, Cambridge (2002)
5. Chaves, M.: Predictive analysis of dynamical systems: combining discrete and continuous formalisms. Ph.D. thesis, Gipsa-lab (2013)
6. Chaves, M., Tournier, L.: Predicting the asymptotic dynamics of large biological networks by interconnections of boolean modules. In: 2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), pp. 3026–3031. IEEE (2011)
7. De Jong, H.: Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol.* **9**(1), 67–103 (2002)
8. Figueiredo, D.: Relating bisimulations with attractors in boolean network models. In: Botón-Fernández, M., Martín-Vide, C., Santander-Jiménez, S., Vega-Rodríguez, M. (eds.) *AICoB 2016*. LNCS, vol. 9702, pp. 17–25. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-38827-4\\_2](https://doi.org/10.1007/978-3-319-38827-4_2)
9. Figueiredo, D., Martins, M.A., Barbosa, L.S.: A note on reactive transitions and reo connectors. In: de Boer, F., Bonsangue, M., Rutten, J. (eds.) *It’s All About Coordination*. LNCS, vol. 10865, pp. 57–67. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-90089-6\\_4](https://doi.org/10.1007/978-3-319-90089-6_4)
10. Gabbay, D., Marcelino, S.: Global view on reactivity: switch graphs and their logics. *Ann. Math. Artif. Intell.* **66**(1–4), 1–32 (2012)
11. Gabbay, D.M.: *Reactive Kripke Semantics*. Cognitive Technologies. Springer, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-41389-6>
12. Gabbay, D.M., Marcelino, S.: Modal logics of reactive frames. *Stud. Logica* **93**(2), 405–446 (2009)
13. Glass, L., Kauffman, S.A.: The logical analysis of continuous, non-linear biochemical control networks. *J. Theor. Biol.* **39**(1), 103–129 (1973)
14. Marcelino, S.R.T.: *Modal logic for changing systems*. University of London (2011)
15. Pola, G., Di Benedetto, M.D., De Santis, E.: Arenas of finite state machines. arXiv preprint [arXiv:1106.0342](https://arxiv.org/abs/1106.0342) (2011)