

Simulating Gene Regulatory Networks using Reaction Systems

Roberto Barbuti, Pasquale Bove, Roberta Gori,
Francesca Levi, and Paolo Milazzo

Dipartimento di Informatica, Università di Pisa, Italy

Abstract. Gene Regulatory Networks represent the interactions among genes regulating the activation of specific cell functionalities and they have been successfully modeled using threshold Boolean networks. In this paper we propose a systematic translation of threshold Boolean networks into Reaction Systems. Our translation produces a non redundant set of rules with the minimal number of objects. This translation allows us to simulate the behavior of a Boolean network simply by executing the (closed) Reaction System we obtain. Moreover, it allows us to investigate the role of different genes simply by “playing” with the rules related to different genes. Starting from a well-known Boolean network model of the Yeast-Cell Cycle, we construct the corresponding Reaction System and start an investigation on causal dependencies among genes.

1 Introduction

In the context of molecular biology of cells, *Gene Regulatory Networks* (GRNs) represent the interactions among genes regulating the activation of specific cell functionalities. More specifically, genes in a GRN can be either active (i.e. the corresponding protein is expressed) or not, and each active gene can either stimulate or inhibit the activation of a number of other genes. Moreover, the activation of some genes is also usually influenced by other factors such as the availability of some substances in the environment or the reception of a signal from neighbor cells. As a consequence, gene regulatory networks can be seen as the mechanism that allow a cell to react to external stimuli. When a stimulus is received, it causes a change in the activation state of a few genes, that, in turn, influence other genes, allowing a new configuration of active genes (corresponding to a new set of active cell functionalities) to be reached.

Several approaches have been proposed to model and analyze GRNs (see [24] for a survey on this topic). Modeling techniques can either deal only with the qualitative aspects of such networks (treating them essentially as logic circuits), or can describe also the quantitative aspects, such as the rates of the interactions. The latter approach is for sure more precise, but requires many additional details of the network dynamics to be taken into account, such as the rates of transcription and translation of genes’ DNA into proteins and the rates of protein-protein and protein-DNA interactions. Qualitative models are often sufficient to reason on the behavior of the regulatory networks, although with some degree of approximation.

In the qualitative modeling setting, one of the most successful modeling frameworks for gene regulatory networks are *Boolean networks*. In this setting, a particular simple form of Boolean networks, the so called *threshold Boolean networks* [16, 19, 22], have been widely used to model the dynamics of quite complex regulatory networks. In threshold Boolean networks, the Boolean function of each node depends on the sum of its input signals only. This variant of Boolean networks can be easily implemented and, at the same time, it is well suited for representing gene regulatory networks.

Boolean networks allow dynamical properties of GRNs to be investigated. Starting from an initial configuration of active genes, the dynamics of a GRNs is expressed as a sequence of steps in which such a configuration is updated according to the influences among the genes described by the Boolean network. Dynamical properties can be investigated either by performing simulations, or by constructing the (finite) graph representing all possible dynamical evolutions. Example of properties that are often studied on these models are reachability and stability of configurations, and confluence of evolutions started from different initial configurations into stable configurations (attractors).

Analysis of dynamical properties may become computationally very expensive. In order to reduce the state space to be analyzed, minimization techniques can be applied. The Boolean function represented by a Boolean network can be synthesized in any framework of logic minimization. The classical approach to logic minimization that produces sum of products two level formulas can be used (see e.g., Espresso [10, 23]). More than two level minimization is harder, but the size of the resulting expressions can significantly decrease. In particular, bounded multilevel forms, such as three or four-level forms [8, 7, 9, 10] could represent a good tradeoff among the cost of the final representation and the time needed for the minimization procedure.

Other analysis methods for GRNs could be applied by changing the representation of the Boolean networks describing them. In this paper we propose a systematic translation of threshold Boolean networks into *Reaction Systems* [17, 13]. Reaction systems were introduced by Ehrenfeucht and Rozenberg as a novel model for the description of biochemical processes driven by the interaction among reactions in living cells. Reaction systems are based on two opposite mechanisms, namely *facilitation* and *inhibition*. Facilitation means that a reaction can occur only if all its reactants are present, while inhibition means that the reaction cannot occur if any of its inhibitors is present. The state of a Reaction System consists of a finite set of objects which can evolve by means of application of reactions. The presence of an object in a state expresses the fact that the corresponding biological entity, in the real system being modeled, is present. Quantities (or concentrations) of the entities are not described: Reaction Systems are hence a *qualitative* modeling formalism.

In this setting the dynamic run of the Reaction System simulates the evolution of the Boolean network. This correspondence allows us to “play” with the rules of the Reaction System related to different genes in order to detect dynamic causality dependencies between genes activation/deactivation. Moreover, we believe that this correspondence will allow us to apply to Boolean networks well-known techniques to detect causality relationships between objects in bio-

logical systems. The understanding of causality relationships among the events happening in a biological (or bio-inspired) system is an issue investigated in the context both of systems biology (see e.g. [18, 11, 12]) and of natural computing (see e.g. [15]).

In [14] Brijder, Ehrenfeucht and Rozenberg initiate an investigation of *causalities* in Reaction Systems [17, 13]. Causalities deal with the ways entities of a Reaction System influence each other. In [14], both static/structural causalities and dynamic causalities are discussed, introducing the idea of *predictor*. In [2, 1, 3, 5, 4, 6], the idea of predictors was enhanced by defining the notions of *formula based predictor* and *specialized formula based predictor*. These new concepts allow us to study all causal dependencies of one object from all others. We believe that the Reaction System simulating a Boolean network can then be investigated by computing the specialized formula based predictor of a particular activation/deactivation gene configuration. This will allow us to obtain a logic formula characterizing all alternative activation/deactivation gene configurations that lead to the requested configuration in a bounded number of steps. This could be very useful to understand which genes are *necessary* for reaching a requested configuration.

The translation we propose in this paper produces a *non redundant* set of rules with the *minimal number* of objects.

The paper is organized as follows. Section 2 introduces the main concepts of (Closed) Reaction Systems. In Section 3 we describe how Boolean networks are defined and how they work. Section 4 presents the systematic translation from Boolean network to Reaction Systems we propose. Finally, in Section 5 we apply our methodology to simulate and study the Yeast-Cell Cycle Boolean Network.

2 Closed Reaction Systems

In this section we recall the basic definition of Reaction Systems [17, 13]. Let S be a finite set of symbols, called objects. A *reaction* is formally a triple (R, I, P) with $R, I, P \subseteq S$, composed of *reactants* R , *inhibitors* I , and *products* P . Reactants and inhibitors $R \cup I$ of a reaction are collectively called *resources* of such a reaction, and we assume them to be disjoint ($R \cap I = \emptyset$), otherwise the reaction would never be applicable. The set of all possible reactions over a set S is denoted by $\text{rac}(S)$. Finally, a *Reaction System* is a pair $\mathcal{A} = (S, A)$, where S is a finite support set, and $A \subseteq \text{rac}(S)$ is a set of reactions.

The state of a Reaction System is described by a set of objects. Let $a = (R_a, I_a, P_a)$ be a reaction and T a set of objects. The result $\text{res}_a(T)$ of the application of a to T is either P_a , if T separates R_a from I_a (i.e. $R_a \subseteq T$ and $I_a \cap T = \emptyset$), or the empty set \emptyset otherwise. The application of multiple reactions at the same time occurs without any competition for the used reactants (threshold supply assumption). Therefore, each reaction which is not inhibited can be applied, and the result of the application of multiple reactions is cumulative. Formally, given a Reaction System $\mathcal{A} = (S, A)$, the result of application of \mathcal{A} to a set $T \subseteq S$ is defined as $\text{res}_{\mathcal{A}}(T) = \text{res}_A(T) = \bigcup_{a \in A} \text{res}_a(T)$.

An important feature of Reaction System is the assumption about the *non-permanency* of objects: the objects carried over to the next step are only those

produced by reactions. All the other objects vanish, even if they are not involved in any reaction.

The dynamics of a Reaction System is generally driven by the *contextual* objects, namely the objects supplied to the system by the external environment at each step. *Closed* Reaction Systems are the subset of general Reaction Systems where the external environment provides objects at the first step only.

This allows us to simplify the dynamics of a (closed) Reaction System $\mathcal{A} = (S, A)$. Indeed, given the initial set D_0 the semantics can be simply defined as the *result sequence*, $\delta = D_1, \dots, D_n$ where each set D_i , for $i \geq 1$, is obtained from the application of reactions \mathcal{A} to the state obtained at the previous step D_{i-1} ; formally $D_i = res_{\mathcal{A}}(D_{i-1})$ for all $1 \leq i < n$. For the sake of simplicity, we write $D_{i-1} \rightarrow_{\mathcal{A}} D_i$ as a shorthand for $D_i = res_{\mathcal{A}}(D_{i-1})$. In this case the sequence of states of the Reaction System coincides with the result sequence $\delta = D_1, \dots, D_n$.

3 Boolean Networks

We present a formal definition of threshold Boolean networks [22] considering a set M of n elements, S_1, S_2, \dots, S_n to be *nodes* of a network. We assign to each element, at each time instant t , a value $S_i(t) \in \{0, 1\}$ denoting if the element S_i is present at that instant or not. The interactions among elements are given by the set of *edges* of the network called E . Each edge in E can be either *activating* or *inhibiting*. An edge from element S_j to element S_i is denoted a_{ij} (where $i \neq j$ given that an element cannot either activate or inhibit itself). An activating edge has value 1 while an inhibiting one has value -1 . Elements M can be partitioned in two sets M_{sa} and M_{nsa} of *self-activating* and *non-self-activating* elements, respectively, $M = M_{sa} \cup M_{nsa}$. A self-activating element if is present at time t and it is not inhibited will be present at time $t + 1$, while a non-self-activating one will not. Moreover we assume that each element S_i has associated a value $\theta_i \in \Theta$ which is called the *threshold parameter* of S_i . The pair (M, E) is called a *threshold Boolean network*.

The states of the nodes in the network are updated in parallel in discrete time. The rules for updating the values of nodes are the following, for $i \in \{1, \dots, n\}$

$$S_i(t+1) = \begin{cases} 1 & \text{if } \sum_j a_{ij} S_j(t) > \theta_i \\ 0 & \text{if } \sum_j a_{ij} S_j(t) < \theta_i \\ S_i(t) & \text{if } S_i \in M_{sa} \wedge \sum_j a_{ij} S_j(t) = \theta_i \\ 0 & \text{if } S_i \in M_{nsa} \wedge \sum_j a_{ij} S_j(t) = \theta_i \end{cases}$$

where the value θ_i is the threshold parameter associated to the element S_i .

Typically the threshold parameter θ_i associated to S_i is equal to 0 so that the switch is inactive if there is no input signal, and it switches on when signals are present. A node which needs more than one incoming signals to be activated can be represented in the model by setting θ_i to a value greater than 0.

Starting from an initial condition, the network produces a dynamical sequence of states and it can reach a periodic attractor or a fixed point.

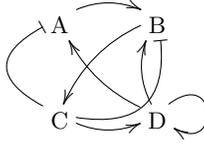


Fig. 1. An example of gene regulatory network.

There is a natural representation of Boolean networks by means of graph where the nodes represent the elements and the edges represent the interactions between the elements; an *activating edge* is indicated by \rightarrow while an *inhibiting* one is indicated by \dashv . Non self-activating genes are represented by nodes with half-arrow (\dashrightarrow) self loops.

We introduce an example to illustrate threshold Boolean networks and their dynamic evolution.

Example 1. Let us consider the threshold Boolean network (M, E) with elements $M = \{A, B, C, D\}$ such that $M_{sa} = \{A, B, C\}$ and $M_{nsa} = \{D\}$ and with the edges depicted in Fig. 1. Thus, the element D is non self-activating while the elements A, B and C are self-activating. We also assume that the threshold parameter for each element is 0.

We describe the temporal evolution considering an initial state in which the element D is present while the others are not. We have

Step	A	B	C	D
1	0	0	0	1
2	1	1	0	0
3	1	1	1	0
4	0	1	1	1
5	0	1	1	0
6	0	0	1	0
7	0	0	1	0

Initially the element D stimulate the activation of both elements A and B because the element C , their inhibitor is not present. Note that at the second step the element D is inactive because it is non self-activating. Then, at step 3, the elements C is present because it is activated by B while A and B are still present because they are self-activating and at step 2 C was not present. At step 4, the element C actives D and inhibits A which is not present anymore. By contrast, C does not inhibit the element B which is still present because it was present at step 3 but also A was present. The step 5 is similar. Finally at step 6 the element B is not present anymore because in the previous configuration A and D were absent and the inhibitor C was activated. The last one is also a stationary state of the system, since no more evolutions of the network are possible from such state.

4 Translation of Boolean Networks into Reaction Systems

We present a translation of threshold boolean networks in closed Reaction System. Given a Boolean network (M, E) with $M = \{S_1, S_2, \dots, S_n\}$ we define for $S_i \in M$,

$$Act(S_i) = \{S_j \mid j \in [1, n] \wedge a_{ij} = 1\} \quad In(S_i) = \{S_j \mid j \in [1, n] \wedge a_{ij} = -1\}$$

We recall that a_{ij} denotes an edge from element S_j to element S_i . Hence, $Act(S_i)$ reports the elements S_j which *activates* S_i and analogously $In(S_i)$ reports the elements S_j which *inhibits* it.

Definition 1. Let (M, E) be a threshold Boolean network with elements $M = \{S_1, S_2, \dots, S_n\}$ and threshold parameters $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$. We define its translation as the closed Reaction System $\mathcal{RS}((M, E)) = (M, A)$, where reactions in A are constructed according to the following inference rules:

$$1) \frac{P_i \subseteq Act(S_i) \quad I_i \subseteq In(S_i) \quad \#P_i - \#(In(S_i) \setminus I_i) = \theta_i - 1}{(P_i, I_i, \{S_i\}) \in A} \quad 2) \frac{S_i \in M_{sa} \quad P_i \subseteq Act(S_i) \quad I_i \subseteq In(S_i) \quad \#P_i - \#(In(S_i) \setminus I_i) = \theta_i}{(P_i \cup \{S_i\}, I_i, \{S_i\}) \in A}$$

The closed Reaction System $\mathcal{RS}((M, E))$ simulates the threshold Boolean network (M, E) using reactions obtained by applying either the inference rule 1) or the inference rule 2). Rule 1) defines the reactions which simulate the production of a element S_i at time $t + 1$ whenever at time t the number of the elements which activate S_i minus the number of the elements which inhibit S_i is greater than θ_i (according to the rule given in Section 3). This behavior is simulated by a reaction which has as *product* S_i , as *reactants* P_i and as *inhibitors* I_i where P_i is a subset of the elements which activates S_i and analogously I_i is a corresponding subset of the elements which inhibits it. Note that this reaction can be applied if in the Reaction System state none of the elements in I_i is present. As a consequence, the set of the elements which are inhibitors of S_i and which may be present is given by $In(S_i) \setminus I_i$. Therefore we guarantee that the cardinality of P_i minus that of $In(S_i) \setminus I_i$ is greater than θ_i . More specifically we require that this difference is equal to $\theta_i - 1$ in order to build a minimal set of reactions in the corresponding Reaction System.

Rule 2) is defined for self-activating elements which remains active at time $t + 1$ if they are present at time t and they are not inhibited (according to the rule given in Section 3). In Reaction System, due to the non-permanency of objects, the objects carried over to the next step are only those produced by reactions. Therefore, in this case, the reaction which simulates the behavior has S_i as *reagent* and also as *product*. Similarly as in the case 1), P_i is a subset of the elements which activates S_i and I_i is a corresponding subset of the elements which inhibits S_i . In this case, however, we require that the cardinality P_i minus the number of inhibitors which might be present (i.e. $(In(S_i) \setminus I_i)$) is exactly θ_i .

Example 2. We give the translation of the threshold Boolean network (M, E) presented in Example 1. Fig. 1) illustrates the interactions between the elements of the network $M = \{A, B, C, D\}$ such that $M_{sa} = \{A, B, C\}$ and $M_{nsa} = \{D\}$.

By assuming again that the threshold parameter for each element is 0 we obtain the closed Reaction System $\mathcal{RS}((M, E)) = (M, A)$ with reactions A are defined as follows:

$$\begin{aligned} & (\{C\}, \emptyset, \{D\}), \quad (\{B\}, \emptyset, \{C\}) \quad (\{C\}, \emptyset, \{C\}), \quad (\{A, D\}, \emptyset, \{A\}) \\ & (\{D\}, \{C\}, \{A\}), (\{A\}, \{C\}, \{B\}) (\{A\}, \{C\}, \{A\}), (\{B, D\}, \emptyset, \{B\}) \\ & (\{D\}, \{C\}, \{B\}), (\{A, D\}, \emptyset, \{B\}) (\{B, A\}, \emptyset, \{B\}), (\{B\}, \{C\}, \{B\}) \end{aligned}$$

The reactions on the left are obtained by applying the rule 1) while those on the right by applying the rule 2). The two columns on the left contains one or more reaction for each elements which produces the element. For the elements D and C there is exactly one reaction given that they do not have inhibitors. By contrast, the element C inhibits both A and B . In the first case, there is just one reaction which says that D produces A if not inhibited by C . The case of B is similar but it can be activated by two element, A and D . Hence, there are three different reactions corresponding to the possible conditions of elements which can activate and inhibit B . Note that the requirements of rule 1) guarantee that only minimal subsets are considered. For instance a reaction such as $(\{A, D\}, \{C\}, \{B\})$ is subsumed by $(\{A, D\}, \emptyset, \{B\})$ given that the latter reaction can be applied regardless of the presence of C .

The two columns on the right shows the reactions for self-activating elements, A , B and C . Similarly as in the previous case there is one or more reaction for each self-activating elements which has the element both as reagent and as product.

We can now prove the soundness of the translation of threshold Boolean networks in closed Reaction System. To relate the configurations of a threshold Boolean network with the state of the associated Reaction System we introduce the following definition.

Definition 2. *Given a threshold Boolean network (M, E) with $M = \{S_1, \dots, S_n\}$, and a state at time t , $S(t) = \{S_1(t), S_2(t), \dots, S_n(t)\}$. The translation of the state $S(t)$ in a corresponding Reaction System state is given by $\mathcal{RS}(S(t))$, defined as follows: $\mathcal{RS}(S(t)) = \{S_i \mid S_i(t) = 1, i \in [1, n]\}$.*

Theorem 1. *Let (M, E) be a threshold Boolean network with elements $M = \{S_1, S_2, \dots, S_n\}$ and threshold parameters $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$. Given a state at time t , $S(t) = \{S_1(t), S_2(t), \dots, S_n(t)\}$ we have that*

$$\mathcal{RS}(S(t+1)) = \text{res}_{\mathcal{A}}(\mathcal{RS}(S(t)))$$

where $\mathcal{A} = \mathcal{RS}((M, E)) = (M, A)$ is the Reaction System obtained by the translation.

Due to Theorem 1 a threshold Boolean network (M, E) with a state $S(0)$ at time 0 can be simulated by the corresponding closed Reaction System $\mathcal{RS}((M, E))$ by considering the initial state $\mathcal{RS}(S(0))$.

At this point, it important to count the number reactions of the closed Reaction System necessary to simulate a threshold Boolean network (M, E) . Such

number depends on the number of the nodes M and of the edges of the network E and also on the threshold parameters Θ . For each $S_i \in M$ the number of the reactions which have S_i as a product depends on the cardinality of $Act(S_i)$ and $In(S_i)$ and on θ_i . Indeed, $Act(S_i)$ and $In(S_i)$ represents the number of the incoming edges which *activates* and *inhibits* S_i respectively.

Proposition 1. *Given a threshold Boolean network (M, E) with elements $M = \{S_1, S_2, \dots, S_n\}$ and threshold parameters $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$. Moreover, let $\mathcal{RS}((M, E)) = (S, A)$ be the corresponding closed Reaction System.*

- For each $i \in \{1, \dots, n\}$, let $N(S_i) = \#\{(R, I, P) \in A \mid S_i \in P\}$ be the number of the reactions which produce the element S_i . We have that

$$N(S_i) = \begin{cases} \sum_{k=1+\theta_i}^{\min(m_i, (l_i+1+\theta_i))} \binom{m_i}{k} \times \binom{l_i}{k-1-\theta_i}, & \text{if } S_i \in M_{nsa}; \\ \sum_{k=1+\theta_i}^{\min(m_i, (l_j+1+\theta_i))} \binom{m_i}{k} \times \binom{l_i}{k-1-\theta_i} + \\ \sum_{h=\theta_i}^{\min(m_i, (l_j+\theta_i))} \binom{m_i}{h} \times \binom{l_i}{k-\theta_i}, & \text{if } S_i \in M_{sa}. \end{cases}$$

where $m_i = \#(Act(S_i))$ and $l_i = \#(IN(S_i))$.

- We have that $\#(A) = \sum_{i=1}^n N(S_i)$.

The previous result is a direct consequence of Definition 1.

Example 3. Let us consider the closed Reaction System $\mathcal{RS}(M, E) = (M, A)$, presented in the Example 2, which is the translation of the threshold Boolean network (M, E) of Example 1. The reaction system has 12 reactions. Indeed, since A, B and C belong to M_{sa} while D belongs to M_{nsa} and the threshold parameter is 0 by applying Proposition 1, we obtain:

$$N(A) = \sum_{i=1}^{\min(1,2)} \binom{1}{i} \times \binom{1}{i-1} + \sum_{i=0}^{\min(1,1)} \binom{1}{i} \times \binom{1}{i} = 1 + (1+1) = 3$$

$$N(B) = \sum_{i=1}^{\min(2,2)} \binom{2}{i} \times \binom{1}{i-1} + \sum_{i=0}^{\min(2,1)} \binom{2}{i} \times \binom{1}{i} = (2+1) + (1+2) = 6$$

$$N(C) = \sum_{i=1}^{\min(1,1)} \binom{1}{i} \times \binom{0}{i-1} + \sum_{i=0}^{\min(1,0)} \binom{1}{i} \times \binom{0}{i} = 1 + 1 = 2$$

$$N(D) = \sum_{i=1}^{\min(1,2)} \binom{1}{i} \times \binom{0}{i-1} = 1$$

5 Simulating the Yeast-Cell Cycle Boolean Network

The cell-cycle process by which a cell goes and divides into two cells is a vital process the regulation of which is conserved among the eukaryotes [21]. The process mainly consists in four phases depicted in Figure 2.

In phase G1 the cell grows and, under appropriate conditions, commits to division, in phase S the DNA is synthesized and chromosomes replicated, G2 is the phase where the cell checks the duplicated chromosomes, and finally in the M (Mitosis) phase the cell is divided into two. After the Mitosis phase, the cell enters the G1 phase, hence completing a cycle. There are about 800 genes

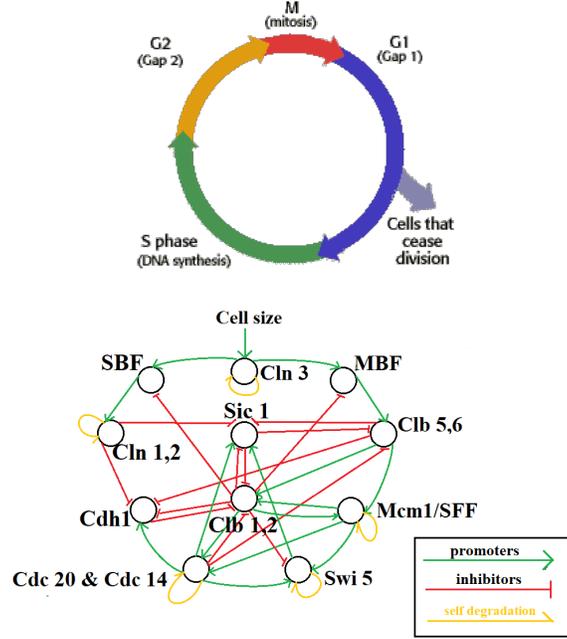


Fig. 3. The Boolean network (M_{Cell}, E_{Cell}) .

involved in the cell-cycle process of the budding yeast [25]. However, the number of key regulators that are responsible for the control and regulation of this complex process is much smaller. Based on extensive literature studies, the authors in [20] constructed a network of key regulators involving 11 genes. The relations between genes are described by the boolean network (M_{Cell}, E_{Cell}) depicted in Figure 3, where the threshold parameter θ is always 0. The boolean network was used to study the time evolution of the protein states. Starting from the $2^{11} = 2048$ possible initial states describing a configuration for gene activation, they discover that all of them flow into one of seven attractor stationary states. In particular, among the seven fixed points there is one big attractor that attracts 1764 initial states. We translated the Boolean network (M_{Cell}, E_{Cell}) of Figure 3 into a Reaction System $\mathcal{A}_{cell} = \mathcal{RS}(M_{Cell}, E_{Cell}) = ((M_{Cell}, A_{Cell})$ by applying the procedure described in Definition 1. The Reaction System \mathcal{A}_{cell} has 52 reactions. For simplicity, we just show the translation of reactions describing the production (activation) of a single node of the Boolean network. Consider the central node named Sic1 in the Boolean network of Figure 3. It has 2 *activating* incoming arcs and 3 *inhibiting* incoming arcs. Since Sic1 $\in M_{sa}$, by Proposition 1 there will be $\sum_{i=1}^{\min(2,4)} \binom{2}{i} \times \binom{3}{i-1} + \sum_{i=0}^{\min(2,3)} \binom{2}{i} \times \binom{3}{i} = (2+3) + (1+6+3) = 15$. Indeed, by applying our translation we find the following rules for the production of Sic1:

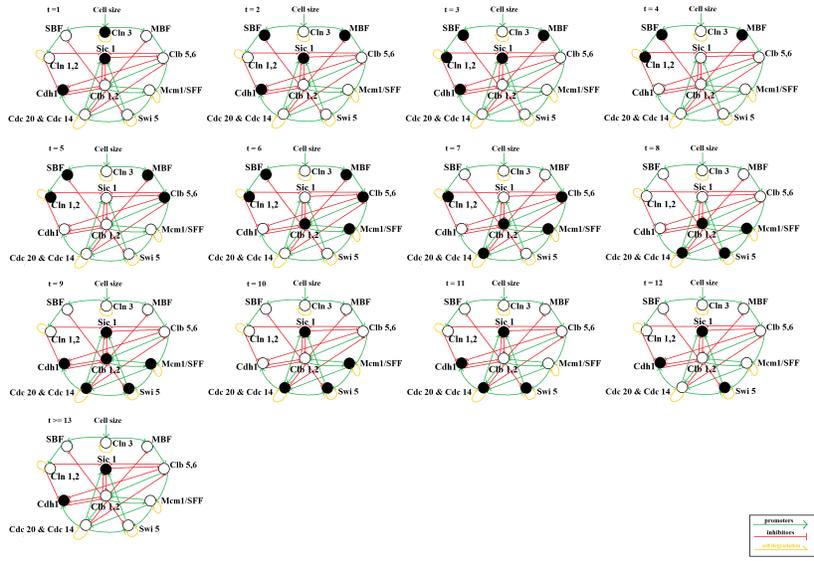


Fig. 4. The cell cycle evolution

$(\{Cdc20\}, \{Clb5, 6, Clb1, 2, Cln1, 2\}, \{Sic1\})$
 $(\{Swi5\}\{Clb5, 6, Clb1, 2, Cln1, 2\}, \{Sic1\})$
 $(\{Cdc20, Swi5\}, \{Clb1, 2, Cln1, 2\}, \{Sic1\})$
 $(\{Cdc20, Swi5\}, \{Clb1, 2, Cln1, 2\}, \{Sic1\})$
 $(\{Cdc20, Swi5\}, \{Cln1, 2, Clb5, 6\}, \{Sic1\})$
 $(\{Sic1\}, \{Clb5, 6, Clb1, 2, Cln1, 2\}, \{Sic1\})$
 $(\{Sic1, Cdc20\}, \{Clb5, 6, Clb1, 2\}, \{Sic1\})$
 $(\{Sic1, Cdc20\}, \{Clb5, 6, Cln1, 2\}, \{Sic1\})$
 $(\{Sic1, Cdc20\}, \{Clb5, 6, Clb1, 2\}, \{Sic1\})$
 $(\{Sic1, Swi5\}, \{Clb5, 6, Clb1, 2\}, \{Sic1\})$
 $(\{Sic1, Swi5\}, \{Clb5, 6, Cln1, 2\}, \{Sic1\})$
 $(\{Sic1, Swi5\}, \{Clb5, 6, Clb1, 2\}, \{Sic1\})$
 $(\{Sic1, Cdc20, Swi5\}, \{Cln1, 2\}, \{Sic1\})$
 $(\{Sic1, Cdc20, Swi5\}, \{Clb1, 2\}, \{Sic1\})$
 $(\{Sic1, Cdc20, Swi5\}, \{Clb5, 6\}, \{Sic1\})$

Note that the behavior of the reactions producing `Sic1` faithfully model the activation of gene `Sic1` in the Boolean network. Consider the case where genes `Cdc20`, `Swi5` and `Clb5, 6` are all active according to the Boolean network of Figure 3 after one step `Sic1` becomes active. The previous state is represented in the Reaction System as the set of activated genes $D_0 = \{Cdc20, Swi5, Clb5, 6\}$. Now starting from D_0 , we can apply rule $(\{Cdc20, Swi5\}, \{Clb1, 2, Cln1, 2\}, \{Sic1\})$ to obtain the production(activation) of gene `Sic1`, therefore $Sic1 \in D_1$ where $D_0 \rightarrow_{\mathcal{A}_{cell}} D_1$. Note that if `Cln1, 2` was also active in the initial state then gene `Sic1` could not be activated according to the Boolean network. This is modeled in the Reaction System by the fact that none of the 15 rules producing `Sic1` could be applied to the set $D_0 = \{Cdc20, Swi5, Clb5, 6, Cln1, 2\}$. Once we have obtained the complete Reaction System \mathcal{A}_{cell} that simulates the entire Boolean network we can run it with any initial state D_0 in order to study the behaviour of the cell when some genes are activated. As a first experiment we executed the Reaction System with the initial state D_0 that it was observed in nature triggers the cell-cycle. Indeed, usually the cell stays in a stationary state where

just genes *Sic1* and *Cdh1* are active. When the cell grows, the external cell size signal *Cell size* arrives and activates *Cln3*. This "excites" the cell from its stationary state and triggers the cycle. We can observe the different states of activation/deactivation of genes during the cell cycle by executing the Reaction System with an initial state where *Sic1*, *Cdh1* and *Cln3* are active. Thus, the following evolution can be observed.

$$\begin{aligned}
& \{Sic1, Cdh1, Cln3\} \xrightarrow{\mathcal{A}_{Cell}} \{SBF, MBF, Sic1\} \xrightarrow{\mathcal{A}_{Cell}} \\
& \{SBF, MBF, Sic1, Cln1, 2\} \xrightarrow{\mathcal{A}_{Cell}} \{SBF, MBF, Cln1, 2\} \xrightarrow{\mathcal{A}_{Cell}} \\
& \{SBF, MBF, Cln1, 2, Clb5, 6\} \xrightarrow{\mathcal{A}_{Cell}} \\
& \{SBF, MBF, Cln1, 2, Clb5, 6, Clb1, 2, Mcm1\} \xrightarrow{\mathcal{A}_{Cell}} \\
& \{Cln1, 2, Clb5, 6, Clb1, 2, Mcm1, Cdc20\} \xrightarrow{\mathcal{A}_{Cell}} \{Clb1, 2, Mcm1, Cdc20, Swi5\} \xrightarrow{\mathcal{A}_{Cell}} \\
& \{Clb1, 2, Mcm1, Cdc20, Swi5, Sic1\} \xrightarrow{\mathcal{A}_{Cell}} \{Mcm1, Cdc20, Swi5, Sic1\} \xrightarrow{\mathcal{A}_{Cell}} \\
& \{Cdc20, Swi5, Sic1, Cdh1\} \xrightarrow{\mathcal{A}_{Cell}} \{Swi5, Sic1, Cdh1\} \xrightarrow{\mathcal{A}_{Cell}} \\
& \{Sic1, Cdh1\} \xrightarrow{\mathcal{A}_{Cell}} \{Sic1, Cdh1\}
\end{aligned}$$

At this point the evolution reaches the stationary state $\{Sic1, Cdh1\}$ and the cell waits for another external stimulus to arrive, that is an external new cell size signal that activates gene *Cln3* and triggers a new cycle. The evolution of the Reaction System represents the evolution of the Boolean network depicted in Figure 4 that describes the entire cell cycle. The Reaction System \mathcal{A}_{Cell} can

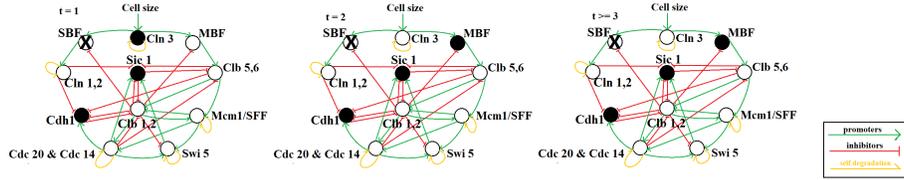


Fig. 5. The cycle evolution where gene *SBF* was silenced

now be used for studying the influence that each gene has in the cell cycle. Each gene can be silenced in turn simply by deleting the rules that produces such gene. Note that this corresponds to simulate the Boolean network where we canceled the node representing the gene together with all his arcs. As a second example consider the case where gene *SBF* was silenced. To this aim, let the Reaction System $\mathcal{A}_{Cell} = (M_{Cell}, A_{Cell})$, we consider the Reaction System $\mathcal{A}_{Cell-SBF} = (M_{Cell}, A_{Cell}/\{(R_a, P_a, \{SBF\}) \mid a \in A_{Cell}\})$. In this case, starting from the stationary state $\{Sic1, Cdh1, Cln3\}$ the following evolution can be observed.

$$\{Sic1, Cdh1, Cln3\} \xrightarrow{\mathcal{A}_{Cell-SBF}} \{Cdh1, MBF, Sic1\} \xrightarrow{\mathcal{A}_{Cell-SBF}} \{Cdh1, MBF, Sic1\}$$

The corresponding evolution on the Boolean network is depicted in Figure 5.

It can be observed that if gene *SBF* was silenced the cell could not perform the cycle because after one step it reaches a new stationary state. This shows

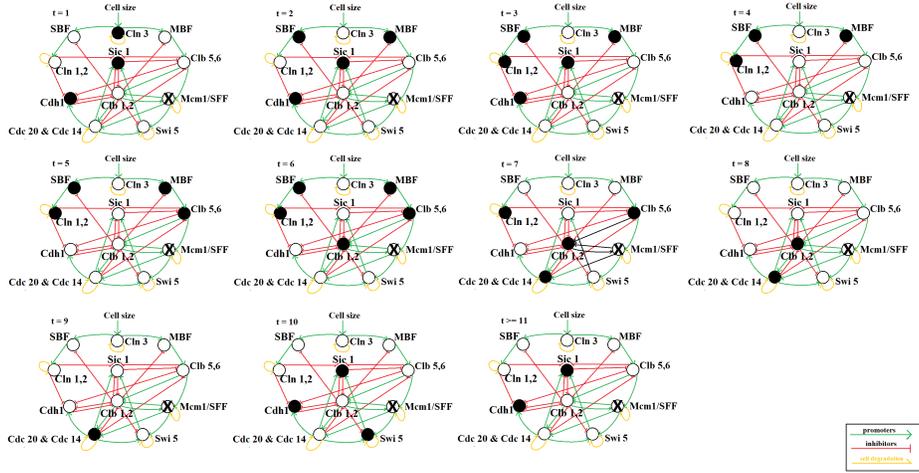


Fig. 6. The cycle evolution where gene *Mcm1* was silenced

that the gene *SBF* was, in some way, necessary for the cycle to be performed. As a last example consider the case where gene *Mcm1* was silenced. The evolution we obtain by considering the Reaction System

$\mathcal{A}_{Cell-Mcm1} = (M_{Cell}, A_{Cell} / \{(R_a, P_a, \{Mcm1\}) \mid a \in A_{Cell}\})$ starting with the stationary state $\{Sic1, Cdh1, Cln3\}$ is directly depicted in Figure 6.

In this case we obtain a very different result from the previous one. Indeed, it can be observed that even if gene *Mcm1* was silenced the cell could perform most of its cycle and go back to the initial stationary state. This suggests that the gene *Mcm1* was not necessary for the cycle to be performed. Indeed, the cell can recover even if, for some reasons, gene *Mcm1* could not be activated.

6 Conclusions

In this paper we proposed a systematic translation of boolean networks into reaction systems. This allows us to simulate the behaviour of a boolean network simply by executing the reaction system we obtain. We applied our method to model the extensively studied Cell cycle that allows a cell to split. The translation of the boolean network describing the Cell cycle into a reaction system allows us to investigate the role of different genes simply by "playing" with the rules related to the production of such gene. In this way, we studied the effects of the silencing of some genes such *SBF* and *Mcm1* on the entire cell cycle.

However we think that the main advantage of our translation will be the application of well known techniques for detecting dynamic causalities relations in reaction systems (see [14, 2, 1, 3, 4]) to determine causality relations between genes of a boolean network.

References

1. Roberto Barbuti, Roberta Gori, Francesca Levi, and Paolo Milazzo. Specialized predictor for reaction systems with context properties. In *Proc. of the 24th Int. Workshop on Concurrency, Specification and Programming, CS&P 2015*, pages 31–43, 2015.
2. Roberto Barbuti, Roberta Gori, Francesca Levi, and Paolo Milazzo. Investigating dynamic causalities in reaction systems. *Theoretical Computer Science*, 623:114–145, 2016.
3. Roberto Barbuti, Roberta Gori, Francesca Levi, and Paolo Milazzo. Specialized predictor for reaction systems with context properties. *Fundamenta Informaticae*, 147(2-3):173–191, 2016.
4. Roberto Barbuti, Roberta Gori, Francesca Levi, and Paolo Milazzo. Generalized contexts for reaction systems: definition and study of dynamic causalities. *Acta Inf.*, 55(3):227–267, 2018.
5. Roberto Barbuti, Roberta Gori, and Paolo Milazzo. Multiset patterns and their application to dynamic causalities in membrane systems. In *Membrane Computing - 18th International Conference, CMC 2017, Bradford, UK, July 25-28, 2017, Revised Selected Papers*, pages 54–73, 2017.
6. Roberto Barbuti, Roberta Gori, and Paolo Milazzo. Predictors for flat membrane systems. *Theor. Comput. Sci.*, 736:79–102, 2018.
7. A. Bernasconi, V. Ciriani, R. Drechsler, and T. Villa. Logic Minimization and Testability of 2-SPP Networks. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 27(7):1190–1202, 2008.
8. A. Bernasconi, V. Ciriani, F. Luccio, and L. Pagli. Synthesis of autosymmetric functions in a new three-level form. *Theory of Computing Systems*, 42(4):450–464, 2008.
9. A. Bernasconi, V. Ciriani, G. Trucco, and T. Villa. *Logic synthesis by signal-driven decomposition*. 2011.
10. Anna Bernasconi, Valentina Ciriani, Gabriella Trucco, and Tiziano Villa. Using Flexibility in P-Circuits by Boolean Relations. *IEEE Trans. Computers*, 64(12):3605–3618, 2015.
11. Chiara Bodei, Roberta Gori, and Francesca Levi. An analysis for causal properties of membrane interactions. *Electr. Notes Theor. Comput. Sci.*, 299:15–31, 2013.
12. Chiara Bodei, Roberta Gori, and Francesca Levi. Causal static analysis for brane calculi. *Theor. Comput. Sci.*, 587:73–103, 2015.
13. Robert Brijder, Andrzej Ehrenfeucht, Michael G. Main, and Grzegorz Rozenberg. A tour of reaction systems. *Int. J. Found. Comput. Sci.*, 22(7):1499–1517, 2011.
14. Robert Brijder, Andrzej Ehrenfeucht, and Grzegorz Rozenberg. A note on causalities in reaction systems. *ECEASST*, 30, 2010.
15. Nadia Busi. Causality in membrane systems. In *Membrane Computing, 8th International Workshop, WMC 2007, Thessaloniki, Greece, June 25-28, 2007 Revised Selected and Invited Papers*, pages 160–171, 2007.
16. Bernard Derrida. Dynamical phase transition in nonsymmetric spin glasses. *Journal of Physics A: Mathematical and General*, 20(11):L721–L725, 1987.
17. Andrzej Ehrenfeucht and Grzegorz Rozenberg. Reaction systems. *Fundamenta Informaticae*, 75(1-4):263–280, 2007.
18. Roberta Gori and Francesca Levi. Abstract interpretation based verification of temporal properties for bioambients. *Inf. Comput.*, 208(8):869–921, 2010.
19. K.E. Kürten. Critical phenomena in model neural networks. *Physics Letters A*, 129(3):157 – 160, 1988.

20. Fangting Li, Tao Long, Ying Lu, Qi Ouyang, and Chao Tang. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences*, 101(14):4781–4786, 2004.
21. Andrew Murray and Tim Hunt. *The Cell Cycle*. Oxford Univ.Press, New York, 1993.
22. Thimo Rohlf and Stefan Bornholdt. Criticality in random threshold networks: annealed approximation and beyond. *Physica A: Statistical Mechanics and its Applications*, 310(1):245 – 259, 2002.
23. R. Rudell and A. Sangiovanni-Vincentelli. Multiple Valued Minimization for PLA Optimization. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 6(5):727–750, 1987.
24. Thomas Schlitt and Alvis Brazma. Current approaches to gene regulatory network modelling. *BMC bioinformatics*, 8(6):S9, 2007.
25. P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown., D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, 9(12):3273–3297, 1998.