

# Towards a theory of modelling with Boolean automata networks – I. Theorisation and observations

Mathilde Noual<sup>1,3</sup> and Sylvain Sené<sup>2,3</sup>

<sup>1</sup> Université de Lyon, ÉNS-Lyon, LIP, CNRS UMR 5668, 69007 Lyon, France

<sup>2</sup> Université d'Évry – Val d'Essonne, IBISC, ÉA 4526, 91000 Évry, France

<sup>3</sup> Institut rhône-alpin des systèmes complexes, IXXI, 69007 Lyon, France

**Abstract.** Although models are built on the basis of some observations of reality, the concepts that derive theoretically from their definitions as well as from their characteristics and properties are not necessarily direct consequences of these initial observations. Indeed, many of them rather follow from chains of theoretical inferences that are only based on the precise model definitions and rely strongly, in addition, on some consequential working hypotheses. Thus, it is important to address the question of which features of a model effectively carry some modelling meaning and which only result from the task of formalising observations of reality into a mathematical language. In this article, we address this question with a theoretical point view that sets our discussion strictly between the two stages of the modelling process that require knowledge of real systems, that is, between the initial stage that chooses a global theoretical framework to build the model and the final stage that exploits its formal predictions by comparing them to the reality that the model was designed to simulate. Taking Boolean automata networks as instances of models of systems observed in reality, we analyse in this setting the remaining stages of the modelling process and we show how the meaning of theoretical concepts can subtly rely on formal choices such as definitions and hypotheses.

*Keywords:* Boolean automata network, update schedule, dynamical behaviour, transition graph, modelling, synchronous and asynchronous transitions.

## 1 Introduction

The manipulation of mathematical objects requires the design and elaboration of precise formal definitions in relation to these objects and to their properties.

Choosing these definitions, specifying consistent connections between them as well as, possibly, restricting and refining them *a posteriori* depends closely on the original purpose of the objects that are considered. Some theoretical objects such as Boolean automata networks may be regarded and studied plainly as mathematical objects that are disconnected from any modelling considerations [1–4]. And they can also be considered as models of other systems, possibly more complex systems such as those encountered and observed in reality [5–8]. Of course, the approaches that follow from these two viewpoints are not independent. On the one hand, formal studies may – and sometimes need to – be fed by practical interrogations that arise from more applicative contexts [9, 10]. On the other hand, by definition, modelling aims at inferring properties of a real system by exploiting a knowledge of the properties of its mathematical model [11, 12]. Yet, with the first point of view, the mathematical objects are studied *per se*. The choices of formalisation and, more generally, the methodology that is adopted mainly aim at building or expanding a purely theoretical understanding in the domain at hand. In particular, possible restrictions to the scope of a study are usually brought for the simple sake of convenience in formal developments. On the contrary, the second point of view introduces motivations of a different nature. Indeed, with the ultimate aim of simulating or explaining a portion of reality, the choices of formalisation become oriented both by a predefined or intuitive interpretation of mathematical notions and by, conversely, the intention of representing given features of reality. As a consequence, the second point of view yields a notable difficulty of which the first purely formal strategy is immune. It requires to navigate safely but constructively between experiences of reality on one side, and mathematical abstractions of it, on the other.

We propose to investigate this central difficulty of the modelling process with a theoretical stance. Thus, considering the second point of view mentioned above, we do not aim at deriving new mathematical results describing the properties of models. Conversely, we do not either address the question of choosing the formal framework to build models for a given category of real systems. Instead, we consider problems in which the general lines of the theory to be used for modelling are assumed to have been set *a priori* and accepted definitely. This way, without needing any thorough knowledge on the systems that are intended to be modelled, we can focus on the *medial* part of the modelling process, *i.e.*, the part that consists in translating reality into abstraction and, dually, confronting abstraction to reality within a predefined formal framework.

In this article, the framework that is considered is that of Boolean automata networks. It serves as a basis to the rest of our discussion. This results in three notable restrictions to the scope of our arguments. First, we consider only the modelling of real systems that are known or supposed to be networks of interacting elements. Second, we suppose that neither the occurrence of events nor the mechanisms that are responsible for them can be observed directly. Only the outcome of events is considered, *i.e.*, only the states (or sequences of states) in which the network elements end up, as a result of unobserved events, are effectively observed. Thirdly, we assume that elements in the networks have two dual

(or possibly extreme) states that can be modelled by theoretical entities, namely *Boolean automata*, that can only take two different states, 0 and 1. In most cases, this last restriction may appear as an excessive oversimplification of reality. If the elements in a network do take more than two states, then, it is probable that the whole range of their different states and the subtle nuances between them impact appreciably on the behaviour of other network elements and, *a fortiori*, on the global network behaviour. Consequently, the system may satisfy properties that are likely to elude a “Boolean modelling” which can only, by nature, focus on the roughest and the most obvious events, such as switches between two extreme states. For this reason, it may be argued that in some cases, a modelling that uses *multi-state* automata rather than Boolean automata is better suited [13–18]. However, we believe that in many cases, precisely because Boolean models can only hope to produce information on very basic, global properties of a system, their explanatory scope may be less ambiguous and more easy to exploit than that of more refined models that account for more subtle and complex properties by relying on a wider range of parameters. Thus, although the knowledge that Boolean modelling helps to develop is much more qualitative than quantitative and certainly very partial, we believe that it potentially provides reliable, well-bounded information or, at least, insights on existing causal relationships, that can serve as solid grounds for further and finer modellings. As mentioned above, however, our objective here is not to discuss the question of the pertinence of a general theoretical framework used to represent a part of reality. Thus, from now on, the Boolean framework is supposed to be accepted and Boolean automata networks are taken to be effective possible satisfactory models of real systems such as, for instance, sets of genes in a cell, interacting via their protein products [19–22].

In the next section, Section 2, we list the main characteristics of these networks that are considered in the literature. This lays the grounds of what we call the “theory of Boolean automata networks”. Then, in Section 3, we analyse the modelling of time and causality. This aims at illustrating some problems that arise within the “theorisation” step of the modelling process which consists in completing the definition of the theory to be used for modelling and simultaneously specifying a correspondence between the modelled features of a category of real systems and the modelling features of the theory. The next section, Section 4, deals with the effective modelling of real systems. It considers the problem of observing the behaviour of a particular real system (belonging to the category of real systems for which the theorisation was intended) and deriving from the resulting, necessarily partial information an understanding of the causes of the events that are observed. Both Sections 3 and 4, in the respective contexts of theorisation and effective modelling, focus on hypotheses that follow naturally and inevitably from choices involved in the formalisation of reality and from the necessity to bypass the incompleteness of the information that is available concerning a certain part of reality. Finally, the last section of this article, Section 5, emphasises the importance of these hypotheses and addresses the more general issue of the ins and outs of modelling with Boolean automata networks.

## 2 Main features of Boolean automata networks

Informally, an *automata network of size  $n$*  involves a set of  $n$  multi-state elements interacting with one another. The elements are called *automata*<sup>4</sup> [4, 23–25]. In the general case, their set of possible states is any (finite) discrete set. In the present particular case of *Boolean automata networks*, they are supposed to take only two possible states, 0 (*inactive*) and 1 (*active*). The interactions between automata of a network consist in influences of some automata states on other automata states. For the sake of simplicity, we abuse language and rather speak of *influences between automata*. When or under what circumstances do these influences effectively produce changes of automata states is a non trivial question that can be answered several ways according to the purpose of the study. The present section aims at completing this rough description and setting up the backbone of the “theory of Boolean automata networks” by enumerating the key features of these networks.

### 2.1 States and configurations

First of all, we introduce some conventions and notations. In the sequel, unless specified otherwise, the Boolean automata networks that are considered are supposed to have size  $n \in \mathbb{N}$  and their automata are assumed to be numbered from 0 to  $n - 1$ . The set  $V = \{0, \dots, n - 1\}$  refers to the set of network automata. As mentioned above, automata are supposed to have only two possible states. The binary set containing these two states is denoted by  $\mathbb{B} = \{0, 1\}$ . Global states of networks, called *configurations* in the sequel, are vectors of the set  $\mathbb{B}^n$ . If  $x = (x_0, \dots, x_{n-1}) \in \mathbb{B}^n$  is the network configuration, then the  $i^{\text{th}}$  component  $x_i \in \mathbb{B}$  of this vector is the *state* of the  $i^{\text{th}}$  network automaton. In our context, focus is put especially on switches of automata states starting in a given network configuration. For this reason, the following notations concerning network configurations will be useful<sup>5</sup>:

$$\begin{aligned} \forall x = (x_0, \dots, x_{n-1}) \in \mathbb{B}^n, \\ (1) \quad \forall i \in V, \bar{x}^i = (x_0, \dots, x_{i-1}, \neg x_i, x_{i+1}, \dots, x_{n-1}), \\ (2) \quad \forall W = W' \uplus \{i\} \subseteq V, \bar{x}^W = \overline{\bar{x}^{W'}} = \overline{\bar{x}^{W'}^i} \text{ and,} \\ (3) \quad \bar{x} = \bar{x}^V = (\neg x_0, \dots, \neg x_{n-1}). \end{aligned} \tag{1}$$

### 2.2 Structure of a Boolean automata network

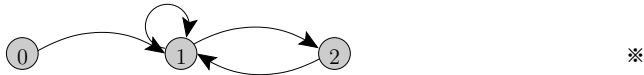
To describe a Boolean automata network, it is often convenient to start by representing its underlying *interaction structure* by a digraph  $\mathcal{G} = (V, A)$ . This

<sup>4</sup> As detailed later, the reason why the term *automata* is used here to refer to network elements is that these elements are supposed to compute output data, that is, their own state, from some given input data, that is, the states of other network automata.

<sup>5</sup>  $\uplus$  denotes the disjoint reunion of sets ( $A = B \uplus C \Leftrightarrow [A = B \cup C \text{ and } B \cap C = \emptyset]$ ) and  $\neg$  denotes the negation of a Boolean value ( $\neg 0 = 1$  and  $\neg 1 = 0$ ).

digraph is called the *interaction graph* of the network. Its set of nodes  $V$  is assimilated to the set of automata of the network and its set of arcs  $A$  represents the set of interactions that take place in it. More precisely, an arc  $(j, i) \in A$  of this digraph represents the influence that (the state of) automaton  $j \in V$  may possibly have on (the state of) automaton  $i \in V$ . Let us note that for the arc  $(j, i)$  to belong to  $A$ , node  $j$  is not supposed to have a *constant* effective impact on  $i$ . It is merely supposed to have an impact in *some* network configurations and in at least one of them (see Equation (2) below). In some works, a digraph  $\mathcal{G}(x) = (V, A(x))$  is defined for every configuration  $x \in \mathbb{B}^n$  [17, 21, 26–28]. It contains arcs  $(j, i) \in A(x)$  such that in  $x$ ,  $j$  does indeed have an appreciable influence on  $i$ . This way, the set of arcs of the interaction graph  $\mathcal{G}$  equals  $A = \bigcup_{x \in \mathbb{B}^n} A(x)$ .

*Example 1.* The following figure represents an interaction graph  $\mathcal{G} = (V, A)$  where  $V = \{0, 1, 2\}$  and  $A = \{(0, 1), (1, 1), (1, 2), (2, 1)\}$ . The network whose structure it represents thus contains three interacting automata. Automaton 0 is influenced by no automaton of the network which means that it always tends to take the same state. Automaton 1 is influenced by all three of the network automata, including itself. And automaton 2 is influenced just by automaton 1.



### 2.3 Local transition functions

The interaction graph of a network gives the existence of the oriented interactions that it involves. However, it does neither specify the nature of these interactions nor the conditions under which they effectively occur. This is done by assigning a *local transition function*  $f_i : \mathbb{B}^n \rightarrow \mathbb{B}$  to each automaton  $i \in V$  so that the following is satisfied:

$$\exists x \in \mathbb{B}^n, f_i(x) \neq f_i(\bar{x}^j) \Leftrightarrow (j, i) \in A. \quad (2)$$

With this new definition, the digraphs  $\mathcal{G}(x)$ ,  $x \in \mathbb{B}^n$  mentioned above contain the following set of arcs:  $A(x) = \{(j, i) \mid f_i(x) \neq f_i(\bar{x}^j)\}$ .

*Example 2.* In agreement with the interaction structure of Example 1, automata 0, 1 and 2 could, for instance, be assigned the following local transition functions, respectively:

$$\forall x \in \mathbb{B}^n, f_0(x) = 1, f_1(x) = x_1 \vee (x_0 \wedge \neg x_2), f_2(x) = \neg x_1. \quad ※$$

### 2.4 Events and updates

In any network configuration, zero, one or several elementary or punctual events may take place. Here, we consider punctual events that consist in the update of one or several automata states. We call them respectively atomic and non-atomic updates and define them formally below.

Supposing that the network is currently in configuration  $x \in \mathbb{B}^n$ , we say that automaton  $i \in V$  is *updated* if its state switches from  $x_i$ , its current state, to  $f_i(x)$ , its new state. Let us note that, possibly,  $f_i(x) = x_i$  so that the update of  $i$  is not effective in  $x$ . In any case, this local event yields a global network configuration change (possibly not effective) which is described by the  *$i$ -update function*  $F_i : \mathbb{B}^n \rightarrow \mathbb{B}^n$ :

$$\forall x \in \mathbb{B}^n, F_i(x) = (x_0, \dots, x_{i-1}, f_i(x), x_{i+1}, \dots, x_{n-1}). \quad (3)$$

This event is said to be *atomic* because it involves only one automaton. We also consider *non-atomic* events that correspond to the simultaneous update of several automata. In the general case, the  *$W$ -update function*<sup>6</sup>  $F_W : \mathbb{B}^n \rightarrow \mathbb{B}^n$  describes the network configuration change that results from the update of all automata in an arbitrary set  $W \subseteq V$ :

$$\forall x \in \mathbb{B}^n, \forall i \in V, F_W(x)_i = \begin{cases} f_i(x) & \text{if } i \in W, \\ x_i & \text{otherwise.} \end{cases} \quad (4)$$

*Example 3.* The following table defines the update functions  $F_1$  and  $F_{\{0,2\}}$  for the network considered in Examples 1 and 2:

$x = (x_0, x_1, x_2)$	$f_0(x)$	$f_1(x)$	$f_2(x)$	$F_1(x)$	$F_{\{0,2\}}(x)$
(0, 0, 0)	1	0	1	(0, 0, 0)	(1, 0, 1)
(0, 0, 1)	1	0	1	(0, 0, 1)	(1, 0, 1)
(0, 1, 0)	1	1	0	(0, 1, 0)	(1, 1, 0)
(0, 1, 1)	1	1	0	(0, 1, 1)	(1, 1, 0)
(1, 0, 0)	1	1	1	(1, 1, 0)	(1, 0, 1)
(1, 0, 1)	1	0	1	(1, 0, 1)	(1, 0, 1)
(1, 1, 0)	1	1	0	(1, 1, 0)	(1, 1, 0)
(1, 1, 1)	1	1	0	(1, 1, 1)	(1, 1, 0)

※

Let us emphasise that the *punctuality* of events mentioned above refers to their happening in a *unique step* whereas the *atomicity* of events characterises their *nature*. All atomic as well as all non-atomic events are punctual. No other punctual events are considered here but the next paragraph mentions more general events consisting in series of successive punctual events.

## 2.5 Transitions and paths

Network *transitions* are couples  $(x, y) \in \mathbb{B}^n \times \mathbb{B}^n$  that represent changes of network configurations (from  $x$  to  $y$ ) due to the occurrence of one or a series of punctual events. Transitions that involve only one punctual event are called *elementary*. They satisfy  $y = F_W(x)$  for some (possibly empty) set  $W \subseteq V$  of automata and are denoted as follows:

$$x \longrightarrow y, x \xrightarrow{W} y \text{ or } x \text{ --- } W \longrightarrow y.$$

<sup>6</sup>  $\forall i \in V$ ,  $F_i$  obviously equals  $F_{\{i\}}$  but in this paper, we prefer the first notation.

There are two main types of elementary transitions. *Asynchronous transitions* correspond to atomic updates. *Synchronous transitions* correspond to non-atomic updates. When emphasis needs to be put on the asynchronicity (resp. synchronicity) of an elementary transition  $x \xrightarrow{\{i\}} y = F_i(x)$  (resp.  $x \xrightarrow{W} y = F_W(x)$ ,  $|W| > 1$ ) it is rather written:

$$x \xrightarrow{i} y, x \xrightarrow{i} y \text{ or } x \xrightarrow{-i} y \text{ (resp. } x \xrightarrow{\blacktriangleright} y, x \xrightarrow{W} y \text{ or } x \xrightarrow{-W} y).$$

General network *transitions*  $(x, y) \in \mathbb{B}^n \times \mathbb{B}^n$  are sequences of zero, one or several elementary transitions. They are denoted using the reflexive and transitive closure  $\xrightarrow{\gg}$  of  $\xrightarrow{\blacktriangleright}$  and defined formally by:

$$x \xrightarrow{\gg} y \Leftrightarrow \exists \ell \in \mathbb{N}, \exists x^1, \dots, x^{\ell-1} \in \mathbb{B}^n, x \xrightarrow{\blacktriangleright} x^1 \xrightarrow{\blacktriangleright} \dots \xrightarrow{\blacktriangleright} x^{\ell-1} \xrightarrow{\blacktriangleright} y. \quad (5)$$

Any network transition  $x \xrightarrow{\gg} y$  thus corresponds to an ordered list of sets  $(W_t)_{1 \leq t \leq \ell}$  such that  $y = F_{W_\ell} \circ \dots \circ F_{W_2} \circ F_{W_1}(x)$ . When this list is known, we use the following notation to specify the sequence of punctual updates in question:

$$x \xrightarrow{W_1, W_2, \dots, W_\ell} y.$$

Now, for the discussion that follows, it is important to note that because transitions involve no other events than automata updates, some situations need to be disregarded. The most basic example is the following:

$$x \xrightarrow{\blacktriangleright} y \text{ where } \exists i \in V, x_i = f_i(x) \neq y_i.$$

Thus, for the network of Examples 1 to 3, the elementary transition

$$(0, 0, 0) \xrightarrow{\blacktriangleright} (1, 1, 0)$$

is impossible although the network can, all the same, perform the non-elementary transition

$$(0, 0, 0) \xrightarrow{\gg} (1, 1, 0)$$

by carrying out the sequence

$$(0, 0, 0) \xrightarrow{\blacktriangleright} (1, 0, 0) \xrightarrow{\blacktriangleright} (1, 1, 0).$$

Similarly, suppose that in configuration  $x$ , two allegedly elementary transitions,  $x \xrightarrow{W} y$  and  $x \xrightarrow{W'} y'$ , are possible. Because any automaton  $i$  that is updated by both transitions necessarily takes state  $f_i(x)$  in both resulting configurations  $y$  and  $y'$ , it must hold that  $\forall i \in W \cap W', y_i = y'_i = f_i(x)$ . Consequently, the following situation is also impossible:

$$x \begin{cases} \xrightarrow{W} y \\ \xrightarrow{W'} y' \end{cases} \text{ where } \exists i \in W \cap W', y_i \neq y'_i.$$

However, again, a similar situation might be possible if the transitions  $x \longrightarrow y$  and  $x \longrightarrow y'$  are not supposed to be elementary. Thus, as discussed further in Section 4, the nature of transitions observed is an essential precision in the observation of the behaviour of a network. To determine it, prior knowledge on the network local transition functions is required.

*Paths* (usually called *trajectories* in the context of dynamical systems) are ordered lists of network transitions  $(x^0, x^1), (x^1, x^2), \dots, (x^{\ell-1}, x^\ell)$ , simply written as follows:

$$x^0 \longrightarrow x^1 \longrightarrow x^2 \longrightarrow \dots \longrightarrow x^{\ell-1} \longrightarrow x^\ell.$$

Since transitions are either elementary or non-elementary, the definition of an arbitrary path may involve steps corresponding to punctual events as well as steps corresponding to undetailed *series* of punctual events (see Example 4). Paths that involve only elementary transitions are said to be *elementary*.

*Example 4.* The path bellow involves two elementary transitions,  $x^0 \longrightarrow x^1 = F_i(x)$  and  $x^2 \longrightarrow x^3 = F_W(x^2)$ , as well as one non-elementary transition,  $x^1 \longrightarrow x^2$ , which could itself be broken into a path of several elementary transitions if the updates it involves were known:

$$x^0 \xrightarrow{i} x^1 \longrightarrow x^2 \xrightarrow{W} x^3. \quad *$$

## 2.6 Update schedules

An *update schedule*  $\delta$  of a set  $V$  of automata (or, by extension, of a network whose set of automata is  $V$ ), is defined by an ordered (finite or infinite) list  $(W_t)_{t \in S}$  ( $S \subseteq \mathbb{N}$ ) of non-empty sets of automata ( $\forall t \in S, \emptyset \neq W_t \subseteq V$ ). We write  $\delta \equiv (W_t)_{t \in S}$  or just  $\delta \equiv W_0, W_1, \dots, W_t, \dots$ . Under an update schedule  $\delta \equiv (W_t)_{t \in S}$ , starting in configuration  $x \in \mathbb{B}^n$ , a network takes sequentially the configurations  $x^0 = F_{W_0}(x)$ ,  $x^1 = F_{W_1} \circ F_{W_0}(x)$ ,  $\dots$ ,  $x^t = F_{W_t} \circ \dots \circ F_{W_0}(x)$ ,  $\dots$ , *i.e.*, it follows the elementary path:

$$x \xrightarrow{W_0} F_{W_0}(x) \xrightarrow{W_1} \dots \xrightarrow{W_t} F_{W_t} \circ \dots \circ F_{W_0}(x) \xrightarrow{W_{t+1}} \dots \quad (6)$$

In particular,  $\delta$  only allows the network to perform elementary transitions that update one of the sets  $W_t$ ,  $t \in S$ . If  $U \subseteq V$  is a set of automata that differs from all of these sets ( $U \neq W_t, \forall t \in S$ ), then  $x \xrightarrow{U} F_U(x)$  is not an elementary transition that can be done under  $\delta$ . In addition, the sets  $W_t$ ,  $t \in S$  themselves cannot either be updated in any network configuration. To detail this, let us introduce by induction on  $t \in S$  the sets  $X_t = \{x \mid x \xrightarrow{W_t} F_{W_t}(x) \text{ is allowed by } \delta\} \subseteq \mathbb{B}^n$ :

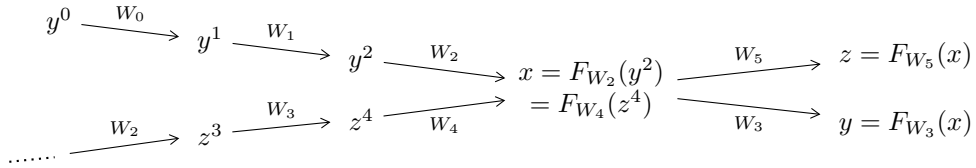
$$\begin{cases} X_0 = \mathbb{B}^n, \\ \forall t \in S, X_{t+1} = \{F_{W_t}(x) \mid x \in X_t\} = F_{W_t}(X_t). \end{cases} \quad (7)$$

Example 5 below illustrates that for any configuration  $x \notin X_t$ , by definition of  $X_t$ , the elementary transition  $x \xrightarrow{W_t} F_{W_t}(x)$  is not possible according to  $\delta$ .



*Example 5.* Suppose that the network considered in Examples 1 to 3 is updated by the periodic update schedule  $\delta \equiv \{1\}, \{0, 2\}, \{1\}, \{0, 2\}, \dots$ . Then, since  $\delta$  does not allow the *atomic* update of automaton 2, any elementary transition of the form  $(0, x_1, x_2) \xrightarrow{\{2\}} (0, x_1, \neg x_1)$  is impossible. The sets  $X_t$ ,  $t \in \mathbb{N}$  defined above can be shown to equal the following:  $X_0 = \mathbb{B}^3$ ,  $X_1 = \mathbb{B}^3 \setminus \{(1, 0, 0)\}$  and,  $\forall t \geq 2$ ,  $X_t = \{(1, 0, 1), (1, 1, 0)\}$ . As a consequence, in configuration  $(1, 0, 0)$ ,  $\delta$  does not either allow the set  $\{0, 2\}$  to be updated.  $\ast$

As mentioned in Section 2.5, our choice of definitions imposes that certain network behaviours be banned. Adding the supplementary constraint of an update schedule restricts further the situations that may be considered possible. For instance, the following situation in which  $x$  belongs to both the sets  $X_3$  and  $X_5$  (see Equation 7 above) is consistent with the update schedule  $\delta$  only if  $\forall i \in W_3 \cap W_5$  it holds that  $y_i = z_i = f_i(x)$ :



and generally, for any subsets  $W_t, W_{t'} \subseteq V$  that belong to the defining list of an update schedule  $\delta$ , the following must hold:

$\forall x \in \mathbb{B}^n$ ,  $[y = F_{W_t}(x) \text{ and } z = F_{W_{t'}}(x)] \Rightarrow [\forall i \in W_t \cap W_{t'}, y_i = z_i = f_i(x)]$ . Thus, again, the set of local transition functions of a network updated with a given update schedule needs to be known in order to determine the trajectories that are possible.

*Periodic update schedules* of arbitrary period  $p \in \mathbb{N}$  correspond to infinite periodic lists  $W_0, W_1, \dots, W_{p-1}, W_0, W_1, \dots, W_{p-1}, \dots$  (*e.g.*, the update schedule of period 2 in Example 5). For the sake of simplicity they are rather defined by finite ordered lists  $(W_t)_{t \in \mathbb{N}/p\mathbb{N}}$  of size  $p$ :  $\delta \equiv W_0, W_1, \dots, W_{p-1}$ . We define *global transition functions*  $F[\delta] : \mathbb{B}^n \rightarrow \mathbb{B}^n$  relative to such update schedules:

$$\forall x \in \mathbb{B}^n, F[\delta](x) = F_{W_{p-1}} \circ \dots \circ F_{W_1} \circ F_{W_0}(x). \quad (8)$$

The definition of this function allows to focus on series of  $p$  elementary transitions rather than on single elementary transitions so that Equation 6 can be simplified to the following (non necessarily elementary) path:

$$x \longrightarrow F[\delta](x) \longrightarrow F[\delta]^2(x) \longrightarrow \dots \longrightarrow F[\delta]^k(x) \longrightarrow \dots,$$

where  $F[\delta]^k$  denotes the  $k^{\text{th}}$  iterate of  $F[\delta]$ .

Now, let  $\approx$  be the equivalence relation that relates periodic update schedules that differ only by a rotation of their sequence of updates. For instance, under this relation, the periodic update schedule defined by the list  $\{1\}, \{0, 2\}, \{1, 2\}$  is equivalent to the two periodic update schedules defined by the lists  $\{0, 2\}, \{1, 2\}, \{1\}$  and  $\{1, 2\}, \{1\}, \{0, 2\}$ . Consider two equivalent update schedules  $\delta \equiv W_0, W_1, \dots, W_{p-1}$  and  $\delta' \equiv W'_0, W'_1, \dots, W'_{p-1}$  satisfying  $\forall t \in \mathbb{N}/p\mathbb{N}$ ,  $W'_t = W_{t+\Delta}$  for some

$\Delta \in \mathbb{N}/p\mathbb{N}$ . Then, let us note that any elementary path starting in  $x \in \mathbb{B}^n$  under  $\delta$  becomes identical, at its  $\Delta^{\text{th}}$  step, to the elementary path that starts in  $F_{W_{\Delta-1}} \circ \dots \circ F_{W_0}(x)$  under  $\delta'$ . Thus, except for  $\Delta$  elementary transitions at the beginning of each path,  $\delta$  and  $\delta'$  yield exactly the same network behaviours. Focusing on non-elementary transitions representing series of  $p$  elementary transitions, however, the two update schedules yield very different paths that cannot generally be identified at all from the point of view of an outside observer that knows nothing about the update schedule that is used.

Let us also note that periodic update schedules  $\delta$  may also be defined as functions  $\delta : V \rightarrow \mathcal{P}(\mathbb{N}/p\mathbb{N})$ , where  $\mathcal{P}(S)$  refers to the power set of a set  $S$  (see Example 6 below). This way, for any automaton  $i \in V$ ,  $\delta(i)$  is the set of updates involving node  $i$  in the periodic sequence  $(W_t)_{t \in \mathbb{N}/p\mathbb{N}}$  such that:

$$\forall t \in \mathbb{N}/p\mathbb{N}, i \in W_t \Leftrightarrow t \in \delta(i).$$

And since each subset  $W_t$ , with  $t \in \mathbb{N}/p\mathbb{N}$ , must be non-empty in order for the update schedule to effectively have period  $p$ ,  $\delta$  must satisfy:

$$\forall t \in \mathbb{N}/p\mathbb{N}, \exists i \in V, t \in \delta(i).$$

A well-known instance of periodic update schedules are *block-sequential update schedules* [4, 23, 29–33]. Their particularity lies in that their sequence of updates involves exactly once each automaton of the network. Thus, they can be defined either by a finite list  $(W_t)_{t \in \mathbb{N}/p\mathbb{N}}$  such that  $V = \bigsqcup_{t \in \mathbb{N}/p\mathbb{N}} W_t$  or, abusing notations introduced above, by a function  $\delta : V \rightarrow \mathbb{N}/p\mathbb{N}$  (see Example 6). The *parallel update schedule* is the unique block-sequential update schedule of period  $p = 1$  ( $\forall i \in V, \delta(i) = 1$ ). It updates all nodes of the network in one step, simultaneously. *Sequential update schedules* are block-sequential update schedules with period equal to the size of the network ( $p = n$ ). They update only one node of the network at a time ( $\forall t \in \mathbb{N}/p\mathbb{N}, |W_t| = 1$ ). Let us mention here that identifying the block-sequential update schedules that are equivalent under  $\approx$  reduces the number of these update schedules <sup>7</sup> by a factor that tends towards  $\frac{2 \cdot \ln 2}{n}$ .

A larger class of periodic update schedules that contains all block-sequential update schedules is the class of *simple update schedules* (mentioned later in Section 2.10 for their particularity). This class contains all periodic update schedules that do not update any automata more than once within each period. With the functional notation introduced earlier on Page 10, simple update schedules are defined as the periodic update schedules satisfying:

$$\forall i \in V, |\delta(i)| \leq 1. \tag{9}$$

---

<sup>7</sup> Proof. Let  $S(n, k)$  for  $k \leq n$  count the number of surjective applications from a set of  $n$  elements to a set of  $k$ . The number of block-sequential update schedules of a set of  $n$  automata equals  $\text{BS}_n = \sum_{0 \leq k < n} \binom{n}{k} \cdot \text{BS}_k = \sum_{1 \leq k \leq n} S(n, k) \sim \frac{1}{2} \cdot \frac{n!}{(\ln 2)^{n+1}}$  [34] (sequence A670 in [35]). The number of equivalence classes for the relation  $\approx$  can be shown to equal  $\widetilde{\text{BS}}_n = \sum_{1 \leq k \leq n} \frac{S(n, k)}{k}$ . Thus, from  $S(n+1, k) = k(S(n, k) + S(n, k-1))$ , results that  $\widetilde{\text{BS}}_{n+1} = 2 \times \text{BS}_n$  and  $\widetilde{\text{BS}}_n \sim \frac{2 \cdot \ln 2}{n} \times \text{BS}_n$ .

Finally, let us introduce another generalisation of block-sequential update schedules. *Fair update schedules* [36, 37] are the periodic update schedules that update each automaton at least once (see Example 6). Unlike block-sequential update schedules, they may update some automata more often than others. A  $k$ -fair update schedule of period  $p$  is an update schedule  $\delta : V \rightarrow \mathcal{P}(\mathbb{N}/p\mathbb{N})$  such that for all automata  $i$  and  $j$  of the network, the following holds:

$$|\delta(i)| \leq k \cdot |\delta(j)|,$$

*i.e.*, within each period,  $i$  is not updated more than  $k$  times as much as  $j$  is. Block-sequential update schedules are a special type of 1-fair update schedule.

*Example 6.* Consider a network of size  $n = 6$ . The 3-fair update schedule  $\delta \equiv \{2, 5\}\{0, 1, 4\}\{1, 2, 3\}\{0, 1, 4, 5\}$ , the block-sequential update schedule  $\beta \equiv \{2\}\{3, 4\}\{0, 1, 5\}$ , the sequential update schedule  $\sigma \equiv \{5\}\{3\}\{1\}\{0\}\{2\}\{4\}$  and the parallel update schedule  $\pi \equiv \{0, 1, 2, 3, 4, 5\}$  can be defined as functions:

$$\delta : \begin{cases} V \rightarrow \mathcal{P}(\mathbb{N}/4\mathbb{N}) \\ 0 \mapsto \{1, 3\} \\ 1 \mapsto \{1, 2, 3\} \\ 2 \mapsto \{0, 2\} \\ 3 \mapsto \{2\} \\ 4 \mapsto \{1, 3\} \\ 5 \mapsto \{0, 3\} \end{cases}, \quad \sigma : \begin{cases} V \rightarrow \mathbb{N}/6\mathbb{N} \\ 0 \mapsto 3 \\ 1 \mapsto 2 \\ 2 \mapsto 4 \\ 3 \mapsto 1 \\ 4 \mapsto 5 \\ 5 \mapsto 0 \end{cases},$$

$$\beta : \begin{cases} V \rightarrow \mathbb{N}/3\mathbb{N} \\ i \in \{0, 1, 5\} \mapsto 2 \\ 2 \mapsto 0 \\ i \in \{3, 4\} \mapsto 1 \end{cases}, \quad \pi : \begin{cases} V \rightarrow \mathbb{N}/1\mathbb{N} \\ \forall i \in V \mapsto 0 \end{cases} \quad *$$

## 2.7 Automata stability and transition effectiveness

As mentioned above, when an automaton is updated in a given configuration  $x \in \mathbb{B}^n$ , it does not necessarily change states. We define the set  $\mathcal{U}(x)$  of automata that can indeed change states in  $x$  and that do if and only if they are updated:

$$\mathcal{U}(x) = \{i \in V \mid f_i(x) \neq x_i\}.$$

The automata in  $\mathcal{U}(x)$  are said to be *unstable* in  $x$  and those in  $\overline{\mathcal{U}}(x) = V \setminus \mathcal{U}(x)$  are said to be *stable* in  $x$ . It is important to note that the only couples  $(x, y) \in \mathbb{B}^n \times \mathbb{B}^n$  that indeed are elementary network transitions are the couples that satisfy:

$$D(x, y) = \{i \in V \mid x_i \neq y_i\} \subseteq \mathcal{U}(x).$$

It also is important to note that for any set  $W \subseteq V$ , the following holds:

$$\forall x \in \mathbb{B}^n, F_W(x) = F_{W \cap \mathcal{U}(x)}(x) = \bar{x}^{W \cap \mathcal{U}(x)}.$$

As a consequence, the elementary transitions

$$x \xrightarrow{W} F_W(x) \text{ and } x \xrightarrow{W \cap \mathcal{U}(x)} F_{W \cap \mathcal{U}(x)}(x)$$

perform different updates but produce identical effects. Elementary transitions  $x \xrightarrow{W} y$  such that  $W \cap \mathcal{U}(x) = \emptyset$  and  $x = y$  are called *null transitions*. Elementary transitions  $x \xrightarrow{W} y$  such that  $W \subseteq \mathcal{U}(x)$  and  $y = \bar{x}^W$  are called *effective transitions*. Other elementary transitions are said to be partially null or partially effective.

## 2.8 State transition systems and transition graphs

A *transition graph* of a network is a digraph  $\mathcal{T} = (X, T)$  that represents its *global behaviour*. Its nodes are network configurations and its arcs are network transitions. There are several important characteristics that a transition graph may have. A first characteristic concerns the nature of the transitions contained in the digraph. If all arcs are elementary transitions, then the transition graph is said to be *elementary*.

Among the set of elementary transition graphs that may be associated to a network of size  $n$  is its *general transition graph* [38], GTG for short, denoted by  $\mathcal{T}_G = (\mathbb{B}^n, T_G)$ . Its set of nodes is  $\mathbb{B}^n$  and its set of arcs is the set of *all* elementary network transitions:

$$T_G = \bigcup \{(x, F_W(x)) \mid x \in \mathbb{B}^n, W \neq \emptyset \subseteq V\}.$$

Every node  $x \in \mathbb{B}^n$  in this multiple digraph has out-degree  $\deg^+(x) = 2^n - 1$ . Another notable elementary transition graph is the *asynchronous transition graph*, or ATG for short. It is the spanning sub-graph  $\mathcal{T}_A = (\mathbb{B}^n, T_A)$  of  $\mathcal{T}_G$  whose set of transitions equals the set of asynchronous transitions of the network:

$$T_A = \bigcup \{(x, F_i(x)) \mid x \in \mathbb{B}^n, i \in V\}.$$

and in which each node  $x \in \mathbb{B}^n$  has out degree  $\deg^+(x) = n$  (the size of the network). Arbitrary transition graphs that contain only asynchronous transitions are also called asynchronous [14, 26, 27, 39–44]. Because updates are not always effective, generally, an elementary transition graph  $\mathcal{T}$  is a multiple digraph. Its simple version is called its *effective version* and is denoted by  $\mathcal{T}^{\text{eff}}$ . In this digraph, all arcs  $(x, y)$ ,  $x \neq y$  which are not loops are effective transitions, *i.e.*,  $D(x, y) \subseteq \mathcal{U}(x)$  (see Example 7 below). The effective version of the GTG is referred to as the  $\text{GTG}^{\text{eff}}$ , and that of the ATG as the  $\text{ATG}^{\text{eff}}$ .

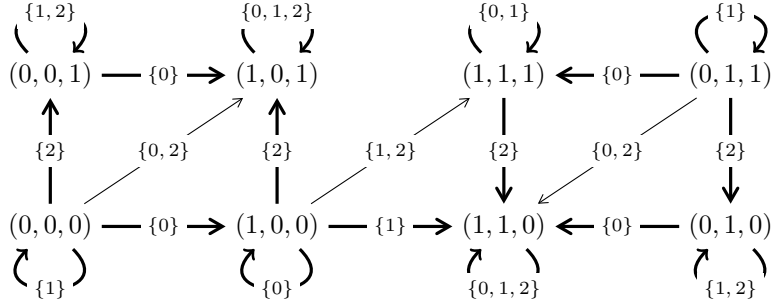
*Non-elementary transition graphs* which contain non-elementary transitions are often used to describe the behaviour of a network that is updated according to a certain periodic update schedule (see Example 7). To a periodic update schedule  $\delta \equiv (W_t)_{t \in \mathbb{N}/p\mathbb{N}}$ , we associate the digraph  $T_\delta$  of  $F[\delta]$  (see Equation 8):

$$\mathcal{T}_\delta = (\mathbb{B}^n, T_\delta) \text{ where } T_\delta = \{(x, F[\delta](x)) \mid x \in \mathbb{B}^n\}. \quad (10)$$

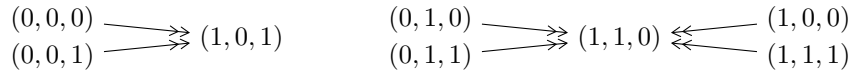
Let us highlight that a transition graph  $\mathcal{T} = (X, T)$  does not necessarily have a node set  $X$  included in  $\mathbb{B}^n$ . Indeed, when the network behaviour is *context-dependent*, *i.e.*, when the transitions that are possible in a given configuration depend on the paths that lead to that configuration, in order to avoid losing any information, it may be required to define  $X$  as a *multi-set* on  $\mathbb{B}^n$ . This is the case for *elementary* transition graphs that describe the behaviour of networks that are updated with periodic update schedules (see  $\mathcal{T}_\delta^{\text{ELEM}}$  in Example 7): if configuration  $x$  belongs to  $k$  different sets  $X_t$  (see Equation 7), then, because the network behaviour is deterministic, any elementary transition graph that represents exhaustively the network behaviour needs to contain  $k$  copies of  $x$  (consider merging nodes representing the same configuration in the digraph  $\mathcal{T}_\delta^{\text{ELEM}}$  of Example 7). Thus, to a periodic update schedule  $\delta \equiv (W_t)_{t \in \mathbb{N}/p\mathbb{N}}$ , we also associate the “elementary version” of  $\mathcal{T}_\delta$ , namely the elementary graph  $\mathcal{T}_\delta^{\text{ELEM}}$  which contains all elementary paths of the network updated with  $\delta$ . Its set of arcs equals set of transitions  $\{(x, F_{W_t}(x)) \in X_t \times X_{t+1}\}$ .

When, on the contrary, the network behaviour at one step is independent of its behaviour at all previous steps, we say that it is *context-free* (or *memory-less*). In this special case, the transition graph  $\mathcal{T} = (X, T)$  defines exactly a *state transition system* [45–47], that is, a digraph whose set of nodes  $X \subseteq \mathbb{B}^n$  is the set of states of the system and whose set of arcs  $T \subseteq X \times X$  is its set of transitions. Contrary to  $\mathcal{T}_\delta^{\text{ELEM}}$ , the transition graphs  $\mathcal{T}_\delta$ ,  $\mathcal{T}_G$ ,  $\mathcal{T}_G^{\text{eff}}$ ,  $\mathcal{T}_A$  and  $\mathcal{T}_A^{\text{eff}}$  mentioned above as well as any of their sub-graphs all are examples of state transition systems.

*Example 7.* Consider the network of Examples 1 to 3. Its GTG<sup>eff</sup> is the digraph represented below where the set of arcs represented with thicker lines equals the set of arcs of the ATG<sup>eff</sup> of the same network.



The transition graph  $\mathcal{T}_\delta$  of the network of Examples 1 to 3 associated to the update schedule given in Example 5 is the following:



and the corresponding elementary transition graph  $\mathcal{T}_\delta^{\text{ELEM}}$  is:

$$\begin{array}{ccccccc}
(0, 0, 0) & \xrightarrow{\{1\}} & (0, 0, 0) & \xrightarrow{\{0, 2\}} & (1, 0, 1) & & \\
(0, 0, 1) & \longrightarrow & (0, 0, 1) & \longrightarrow & (1, 0, 1) & & \\
(0, 1, 0) & \longrightarrow & (0, 1, 0) & \longrightarrow & (1, 1, 0) & & \\
(0, 1, 1) & \longrightarrow & (0, 1, 1) & \longrightarrow & (1, 1, 0) & & \\
\\
(1, 0, 0) & \xrightarrow{\{1\}} & (1, 1, 0) & \xrightarrow{\{0, 2\}} & (1, 1, 0) & & \\
(1, 0, 1) & \longrightarrow & (1, 0, 1) & \longrightarrow & (1, 0, 1) & & \\
(1, 1, 0) & \longrightarrow & (1, 1, 0) & \longrightarrow & (1, 1, 0) & & \\
(1, 1, 1) & \longrightarrow & (1, 1, 1) & \longrightarrow & (1, 1, 0) & & \ast
\end{array}$$

## 2.9 Network behaviours

If  $\mathcal{T} = (X, T)$  is the transition graph representing the *global* behaviour of a given network, then, any sub-graph of  $\mathcal{T}$  represents a *particular* behaviour of that network. For instance, if a network is supposed to be potentially able of performing *any* elementary transition, then,  $\mathcal{T}$  either equals the GTG  $\mathcal{T}_G$  of the network or its GTG<sup>eff</sup>  $\mathcal{T}_G^{\text{eff}}$  and any elementary transition graph represents a possible particular behaviour of the network.

Let us focus on state transition systems  $\mathcal{T} = (\mathbb{B}^n, T)$ . For these, we define the binary relation  $T^* \subseteq \longrightarrow$  as the reflexive and transitive closure of the relation  $T$ , that is,  $(x, y) \in T^*$  if and only if there exists in  $\mathcal{T}$  a path from  $x$  to  $y$ . *Transient configurations* are then defined as the configurations  $x \in \mathbb{B}^n$  that satisfy:

$$\exists y \in \mathbb{B}^n, (x, y) \in T^* \wedge (y, x) \notin T^*.$$

Any behaviour of the network that involves transient configurations is said to be transient itself. Configurations that are not transient are called *recurrent*. These configurations are precisely those that induce the terminal strongly connected components of  $\mathcal{T}$ , called *limit behaviours* here (and rather called *attractors* in the context of deterministic dynamical systems [48–50]). There are two main types of limit behaviours. Those that contain strictly more than one configuration are called (sustained) *oscillations* [51] (or *limit cycles* when  $\mathcal{T}$  defines a deterministic dynamical system). Limit behaviours of size one are called *stable configurations* (or *fixed points* when  $\mathcal{T}$  defines a deterministic dynamical system<sup>8</sup>). They are characterised by their out-going effective degrees<sup>9</sup> being equal to 0. Note that in a stable configuration  $x$ , it is not necessary that all automata of the network be stable (*i.e.*, that  $\mathcal{U}(x) = \{i \in V \mid f_i(x) \neq x_i\}$  be empty). Indeed, for  $x$  to be stable, it suffices that  $\mathcal{T}$  contains no transition that corresponds to the update in  $x$  of automata belonging to  $\mathcal{U}(x)$  (this happens in particular when  $\mathcal{T} = \mathcal{T}_\delta$ ).

<sup>8</sup> In this case,  $\mathcal{T}$  is the graph of a global transition function  $F : \mathbb{B}^n \rightarrow \mathbb{B}^n$  so any stable configuration in this graph is a fixed point of that function.

<sup>9</sup> In any transition graph  $\mathcal{T}$ , the *out-going effective degree* of configuration  $x$  is the number of arcs out-going  $x$  that are *effective* transitions.

## 2.10 Relationships between network features

In the sequel, we informally call *observer of a network* “anyone” that has full, partial or no knowledge of its characteristics (structure, set of interactions, behaviour. . .). In this section, we discuss how the different features of the network relate and how an observer carrying one type of information on the network may derive additional information of another type.

First, the information carried by the interaction graph (as defined in Sections 2.2 and 2.3) of a network is contained in the information carried by its set of local transition functions. However, it takes exponential time in the size  $n$  of the network to draw an interaction graph from a set of local transition functions. This complexity stands even when the local transition functions are given in conjunctive normal form (CNF). Indeed, given  $j \in V$  and the CNF definition of  $f_i$ , the problem of determining whether there exists  $x \in \mathbb{B}^n$  such that  $f_i(x) \neq f_i(\bar{x}^j)$  is NP-complete: the CNF version of SAT [52, 53] which is NP-complete can be reduced to it.

Next, let us emphasise that with the sole knowledge of the set of local transition functions of a network, many different transition graphs may be built. To choose one of these graphs as representing the actual network behaviour therefore requires additional information. This additional information may simply be, for instance, the datum specifying that all elementary network transitions are possible, or at least that no elementary network transitions are known or considered to be impossible. In this case, either the GTG or the GTG<sup>eff</sup> (which can both be built in time  $\mathcal{O}(n \cdot 2^{2n})$ ) need to be chosen. They are the transition graphs that represent the alleged network behaviour the most completely. If the network is known to perform only asynchronous transitions, then the ATG, the ATG<sup>eff</sup> (which can both be built in time  $\mathcal{O}(n \cdot 2^n)$ ) or sub-graphs of these are better suited. And for networks that are supposed to be updated with a given update schedule  $\delta$  of period  $p$ , either  $\mathcal{T}_\delta^{\text{elem}}$  or  $\mathcal{T}_\delta$  (which can both be built in time  $\mathcal{O}(np \cdot 2^n)$ ), must be considered. Thus, in short, in addition to the knowledge of the set of local transition functions, to derive the transition graph of a network, it is necessary (but not always sufficient) to specify the *nature* of the possible transitions or paths of the network. Without any such indication, the network behaviour cannot be inferred non-ambiguously even from an exhaustive knowledge of its underlying mechanisms (*i.e.*, its structure and interactions).

Conversely, knowing the transition graph of the network is not sufficient to infer its set of local transition functions and its structure [4, 23, 30, 54]. In some cases, however, with some simple additional information, it is [55, 56]. Indeed, first, suppose that the network behaviour is known to be fully described by a *deterministic* state transition system  $\mathcal{T} = (\mathbb{B}^n, T)$  in which all nodes have out-degree at most 1 and all transitions are of the form  $(x, F(x))$  for a certain global transition function  $F : \mathbb{B}^n \rightarrow \mathbb{B}^n$ . Then, from Equation 11 below, the set of local transition functions  $\mathcal{F} = \{f_i \mid i \in V\}$  of the network can be derived in time

$\mathcal{O}(n \cdot 2^n)$ , *i.e.*, in linear time with respect to the size of  $\mathcal{T}$  [30].

$$\forall i \in V, f_i : \begin{cases} \mathbb{B}^n \rightarrow \mathbb{B} \\ x \mapsto F(x)_i \end{cases}. \quad (11)$$

If  $\mathcal{T}$  is known to be a sub-graph of the ATG or the ATG<sup>eff</sup>, then, also in linear time with respect to the size of  $\mathcal{T}$  ( $\mathcal{O}(n^2 \cdot 2^n)$ ),  $\mathcal{F}$  can be built using:

$$\forall i \in V, f_i : \begin{cases} \mathbb{B}^n \rightarrow \mathbb{B} \\ x \mapsto \begin{cases} \neg x_i & \text{if } (x, \bar{x}^i) \in T, \\ x_i & \text{otherwise.} \end{cases} \end{cases} \quad (12)$$

More generally, under the hypothesis that *all transitions in  $\mathcal{T}$  are elementary*, the set  $\mathcal{F}$  of local transition functions of the network can be derived in time  $\mathcal{O}(n \cdot 2^{2n})$  by exploiting the following equation:

$$\forall i \in V, f_i : \begin{cases} \mathbb{B}^n \rightarrow \mathbb{B} \\ x \mapsto \begin{cases} \neg x_i & \text{if } \exists y \in \mathbb{B}^n, (x, y) \in T \text{ and } y_i \neq x_i, \\ x_i & \text{otherwise.} \end{cases} \end{cases} \quad (13)$$

In particular, Equation 13 can be used when the network is supposed to be updated in parallel. For other block-sequential and simple update schedules, Algorithm 1 builds a set of local transition functions  $\mathcal{F}$  from the dual input information of a transition graph  $\mathcal{T}$  and an update schedule  $\delta$ . To do so, it requires that  $\mathcal{T}$  be indeed the graph of a function  $F : \mathbb{B}^n \rightarrow \mathbb{B}^n$  so that  $\mathcal{T} = \mathcal{T}_\delta$  and  $F = F[\delta]$  be possible. In addition, importantly,  $\delta$  must be a simple update schedule. The reason for this restriction is that Algorithm 1 (just as an observer would need to do) relies on the knowledge that every apparent behaviour has no hidden cause. If the transition  $x \longrightarrow y$  is observed, then, on the one hand, all automata that have not *apparently* changed states ( $y_i = x_i$ ) have not *effectively* done so. If they had, they would have had to change an even number of times in order to come back eventually into their initial state ( $x_i$ ). This is not possible when automata are updated at most once between two observations of the network configuration. On the other hand, if the automata that do change states have not done so as a result of a series of unobservable causes, then their changes of states can indeed be exploited to derive the interactions which are their direct causes.

Let us highlight that block-sequential update schedules have the notable particularity of allowing to observe the network configuration only once per period of updates without loosing any crucial information about the nature of limit behaviours of the network. It suffices to know the usually non-elementary<sup>10</sup> transition graph  $\mathcal{T}_\delta$  to draw some significant information. For example, consider a network updated with a block-sequential update schedule  $\delta \equiv (W_t)_{t \in \mathbb{N}/p\mathbb{N}}$  of period  $p$ .  $\mathcal{T}_\delta$  necessarily contains a sub-graph of the following form, that is, a limit

<sup>10</sup> For update schedules  $\delta$  such as those that are considered in the present paragraph, unless the period of  $\delta$  is 1 as for the parallel update schedule,  $\mathcal{T}_\delta$  is non-elementary.





for the modelling of a certain category of real systems such as systems of genes that interact via their protein products. In other words, it aims at setting some grounds before any practical modelling of a real system (this next step frames Section 4) can effectively be done. Theorisation thus first needs to choose a formal language in order to describe the features of “reality” that are considered<sup>11</sup>. In our context, this language consists in the mathematical language that allows to express definitions relative to Boolean automata networks. Next, or simultaneously, in this language, a *theory* is defined. In consistency with one another, definitions are given to formal objects. Properties of these and relationships between them are specified. This leads, for instance, to the theory of Boolean automata networks as described in Section 2. From the language and theory that are chosen follows immediately a correspondence between features of reality intended to be modelled and features of the theory supposed to model them. This reality/theory correspondence is dually composed of a *modelling map* (which, informally represents the *reality*  $\rightarrow$  *theory* direction of the correspondence) and an *interpretation map* (representing the opposite direction, *theory*  $\rightarrow$  *reality*). The *modelling map* specifies how portions of reality are represented mathematically. For example, it may specify that genes are modelled by automata, that interactions between genes via the proteins they code for are represented by Boolean functions, that changes of protein concentrations in the cell are simulated by transitions of a transition graph. . . Conversely, the *interpretation map* associates a “modelling meaning” or a justification (possibly void) to each object and property of the theory. For example, according to this map, automata may be interpreted as genes, Boolean vectors may be interpreted as cell configurations and a local transition function  $f_1 : x \mapsto \neg x_0 \wedge x_1$  can be regarded as modelling the fact that gene  $G_1$  remains expressed only if gene  $G_0$  is not, given that automaton  $i \in \{0, 1\}$  models gene  $G_i$ . Further, the “interpretation map” can also specify that a formal hypothesis stating that synchronous transitions are impossible (see Hypothesis 1 below) is the formal translation of the fortuitousness of two events ending simultaneously.

The main difficulty of the theorisation step of modelling is the definition of both the modelling and the interpretation maps. Indeed, from the deliberately designed backbone of the reality/theory correspondence, it often follows some complex and subtle ramifications. These ramifications can impose implicitly that certain features of reality be matched to some precise features of the theory and conversely. But they can also, on the contrary, forbid some matches. Further, generally, neither the modelling map nor the interpretation map, are surjective: not all aspects of the theory can necessarily be interpreted as the representation of something real and, obviously, not all aspects of reality are represented

---

<sup>11</sup> Let us note that the very act of describing mentally some observations is already in itself an automatic and often subconscious theorisation of reality so there obviously are some theorisations occurring at a “lower-level” than the level which is the object of this section. However, we ignore them here because their identification lies outside the scope of our competences (it requires, in particular, to have a good idea of what “reality” is precisely).

by the theory. As a consequence, the modelling framework itself requires thorough coherence and bounding in its definition. *Modelled features* of reality need to be identified and distinguished from its *non-modelled features*. Dually, *modelling parameters* and *properties* of a theory, which can reasonably be considered as representations of some portion of observed reality, also need to be identified and distinguished from its *non-modelling features* (those that are artefacts of formalisation rather than pertinent representations of anything real). The present section focuses on the notion of time to highlight these difficulties and the coherence that is required between the distinct associations specified or implied by the reality/theory correspondence.

The very concept of *transition* from one network state  $x$  to another  $y$  suggests a notion of time that positions  $x$  *before*  $y$ . The term *trajectory* that is usually used instead of *network path* (see Section 2.5) re-enforces the natural association that can be made between an intuitive idea of time flow and a mathematical concept of causal precedence. The *length* of a series of transitions from  $x$  to  $y$  evokes the *time* that the system spends to go from one point  $x$  to another  $y$  in its state space. And, especially in a modelling context where the aim precisely is to relate experiences of reality and abstract concepts, the formal language used tends to adapt to implicit associations that are made to better understand theoretical objects and their properties. The question of the pertinence of these associations, however, is not always obvious. In the sequel, we first recall some definitions on dynamical systems and show how Boolean automata networks with a given behaviour can be seen in terms of this formalism. Then, we discuss how, in different formalisms, different points of view on the way time is taken into account in a model give rise to different problems and questions in its theoretical analysis. The section ends with an example that illustrates some of these questions and in which Boolean automata networks are seen as models of genetic regulation networks.

### 3.1 Dynamical systems

A discrete-time dynamical system, called simply *dynamical system* in the sequel<sup>12</sup> is a triplet  $\mathcal{D} = (S, \Theta, \phi)$  where  $S$  is the *state space* of the system,  $\Theta \subseteq \mathbb{N}$  is its *time domain* (or *evolution space*) and  $\phi : S \times \Theta \rightarrow S$  is the *evolution function* describing the system dynamics. It satisfies:

$$\forall s \in S, \forall t_1, t_2 \in \Theta, \phi(s, 0) = s \text{ and } \phi(\phi(s, t_1), t_2) = \phi(s, t_1 + t_2).$$

$\phi(s, t)$  represents the state of the system at time  $t$  so that the *trajectory* (or *path*) of  $\mathcal{D}$  initiated in state  $s \in S$  is  $\{\phi(s, t) \mid t \in \Theta\}$ . Let us note that for any initial state  $s \in S$ , the function  $\phi_s : t \mapsto \phi(s, t)$  associates to every time step  $t$ , a *unique* image  $\phi(s, t)$ . In particular, it defines the unique successor  $\phi(s, 1)$  of  $s$ .

<sup>12</sup> We bypass the difficulty of choosing a time space  $\Theta$  by assuming it is discrete. Continuous-time dynamical systems will thus not be mentioned at all. This choice is supported by the state space being discrete in our framework.

This allows for two types of dynamical systems: deterministic and stochastic. In the case of stochastic dynamical systems, however, the terminology introduced above needs to be adapted so that  $S$  rather denotes *a set of probability laws* on the state space of the system:

$$S \subseteq \{\mu \in [0, 1]^{2^n} \mid \sum_{x \in X} \mu_x = 1\}.$$

In this case, let  $x(t)$  be the random variable corresponding to the system state at time  $t$  and let  $\mu(t)$  be the probability law of  $x(t)$ . Then,  $\phi$  is such that  $\mu(t) = \phi(\mu(0), t)$ .

**Deterministic dynamical systems** As an example of deterministic dynamical systems, let us consider a network updated with a periodic update schedule  $\delta \equiv (W_k)_{k \in \mathbb{N}/p\mathbb{N}}$ . As discussed in Section 2.8, there are several ways to describe the network behaviour in this case. First, focusing globally on *periods* of  $\delta$ , one may consider the transition graph  $\mathcal{T}_\delta$  in which every configuration has out-degree 1. The system is then context-free and can be defined as a deterministic dynamical system  $\mathcal{D} = (\mathbb{B}^n, \mathbb{N}, \phi)$  where:

$$\forall x \in \mathbb{B}^n, \forall t \in \mathbb{N}, \phi(x, t) = F[\delta]^t(x).$$

A second way to describe the network behaviour under  $\delta$  is to decompose the transitions  $x \longrightarrow F[\delta](x)$  into series of elementary transitions

$$x \xrightarrow{W_t} F_{W_t}(x), \quad t \in \mathbb{N}/p\mathbb{N}.$$

This yields the transition graph  $\mathcal{T}_\delta^{\text{elem}}$  which is not a state transition system like  $\mathcal{T}_\delta$  but in which every node still has out-degree 1.  $\mathcal{T}_\delta^{\text{elem}}$  defines a context-dependent system (whether or not  $W_0$  can be updated in configuration  $x$ , for instance, depends on the previous elementary transition performed by the system) that can be seen as a deterministic dynamical system  $(\mathbb{B}^n, \mathbb{N}, \phi)$  where:

$$\forall x \in \mathbb{B}^n, \forall t = k \cdot p + d \equiv d \pmod{p}, \phi(x, t) = F_{W_d} \circ \dots \circ F_{W_1} \circ F_{W_0} \circ F[\delta]^k.$$

In this case, as the definition of  $\phi$  shows, there is no time-independent (global transition) function  $\phi_1$  that associates a unique successor  $\phi_1(x) = \phi(x, 1)$  to every configuration  $x$ .

**Stochastic dynamical systems** When the transition graph  $\mathcal{T} = (\mathbb{B}^n, T)$  describing the behaviour of a network has nodes of out-degree greater than 1, there generally is no obvious way of defining it as a dynamical system. However, with some additional indications or hypotheses, probabilities may be assigned to each transition of  $\mathcal{T}$ . This way,  $\mathcal{T}$  can be seen as the graph of a Markov chain on  $\mathbb{B}^n$  and the adjacency matrix of  $\mathcal{T}$  can be turned into a *stochastic transition matrix*, often called the Markovian matrix, *i.e.*, a matrix  $P$  of dimension  $2^n \times 2^n$

satisfying:

- (i)  $\forall x, y \in \mathbb{B}^n, P_{x,y} \in [0, 1]$ ,
- (ii)  $\forall x \in \mathbb{B}^n, \sum_{y \in \mathbb{B}^n} P_{x,y} = 1$  and
- (iii)  $\forall x, y \in \mathbb{B}^n, (x, y) \notin T \Rightarrow P_{x,y} = 0$ .

The component  $P_{x,y}$  in this matrix represents the probability that the network performs transition  $(x, y)$ , *i.e.*, the probability that it reaches configuration  $y$  given that it was in configuration  $x$  at the previous time step:

$$P_{x,y} = \mathbb{P}(x(t+1) = y \mid x(t) = x).$$

The probability that it changes configurations is given by:

$$\mathbb{P}(x(t+1) \neq x(t)) = \sum_{y \neq x \in \mathbb{B}^n} P_{x,y}.$$

Then, the network behaviour can be defined as a stochastic dynamical system whose evolution function is given by:

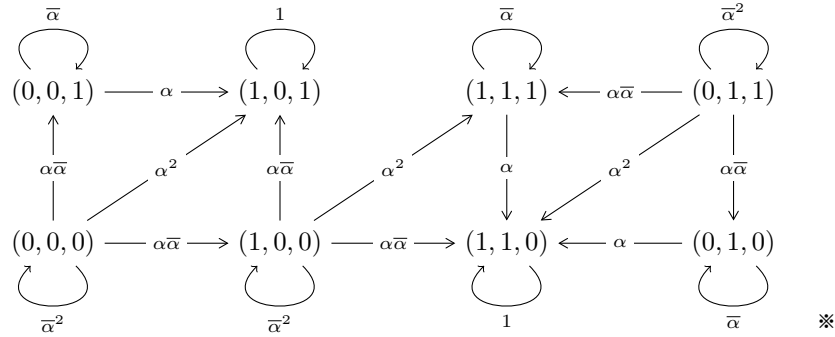
$$\begin{cases} \phi(\mu, 0) = \mu, \\ \phi(\mu, t) = \mu \cdot P^t, \end{cases}$$

where  $\mu$  is an arbitrary probability law on the state space of the system ( $\mu \in [0, 1]^{2^n}$  and  $\sum_{x \in X} \mu_x = 1$ ). Thus, if  $\mu = \mu(0)$  is the probability law of the initial network configuration  $x(0) \in \mathbb{B}^n$ , then  $\mu(t) = \phi(\mu, t)$  is the probability law of the network configuration  $x(t) \in \mathbb{B}^n$  at time step  $t$  ( $\forall x \in \mathbb{B}^n, \mu_x(t) = \mathbb{P}(x(t) = x)$ ).

*Example 8.* Consider the network of Examples 1 to 3 and its  $\text{GTG}^{\text{eff}} \mathcal{T}_G^{\text{eff}} = (\mathbb{B}^3, T)$  and given in Example 7. Then, introducing a rate  $\alpha \in [0, 1]$  at which automata of the network are updated (at each time step, any automaton  $i \in V$  is updated with probability  $\alpha$ ) [57–59], the stochastic transition matrix  $P$  can be defined as follows:

$$\forall (x, y) \in \mathbb{B}^3 \times \mathbb{B}^3, P_{x,y} = \begin{cases} \alpha^{d_{xy}} \cdot (1 - \alpha)^{u_x - d_{xy}} & \text{if } (x, y) \in T, \\ 0 & \text{otherwise,} \end{cases}$$

where  $d_{xy} = |D(x, y)|$  and  $u_x = |\mathcal{U}(x)|$  (see Section 2.7 for notations). This yields the following transition graph where transitions of  $\mathcal{T}_G^{\text{eff}}$  are labelled by their probabilities:



### 3.2 Modelling time

In the case of networks seen as dynamical systems (deterministic or stochastic), because the set  $\Theta$  is called the time domain of the system and  $t \in \Theta$  is called a time step, a *moment* or a *date*, an implicit notion of time is introduced that can be interpreted several ways. In this section we discuss three different points of view that can be taken, in the context of modelling, on the abstractions of time arising from the definition of a network behaviour. The first two points of view derive from the formalisms of dynamical systems. The last one follows from that of state transition systems which are called “*causal systems*” here to emphasise their difference with dynamical systems<sup>13</sup>.

**Modelling durations** A first reading of the mathematical concept of time defined by dynamical systems consists in interpreting it as a literal match of the real time so that time steps in  $\Theta$  are taken as a unit of measurement of real time and all possible network transitions are supposed to take the same amount of time, that is, one unit. When the network behaviour is described by an elementary transition graph  $\mathcal{T}$ , for instance, all elementary updates are supposed to take the same time, whatever the automata that they update.

When the system movements modelled by transitions can however not all be assumed to take the same fixed amount of time,  $\Theta$  cannot be interpreted as a discretised version of a real time flow. In that case, to maintain a modelling of transition durations, the time domain must be backed up with some additional parameters. One straightforward method is to label each transition of the network by a value that measures the time taken by this transition, or rather, by the event modelled by this transition. In these lines, in [8, 42, 60–63], the authors have refined the formalism of Boolean automata networks in order to model genetic regulation networks and take into account some of the time delays to which regulations are submitted (see Section 3.3).

In addition to the questions that are mentioned in the next paragraphs which are also natural and pertinent with more general approaches, this point of view (like any point of view) on time yields a set of theoretical questions that are specific to it. These questions rely on the strong hypothesis that the concept of time in the definition of a dynamical system is indeed meaningful in terms of modelling. As an example, let us cite the following non-exhaustive list: *How long does the network take or is the network expected to take to reach a certain configuration or to start displaying a certain behaviour? What is the (most likely) network configuration that is reached in time  $t$ ? When or how long will the network display this behaviour?, How many times is the network expected to reach this configuration during this lapse of time?...*

---

<sup>13</sup> Let us note that for similar reasons, that is, to emphasise the difference between the two main angles (causal and dynamical) that can be adopted to study Boolean automata networks, in Section 2, we have deliberately chosen to use the term *behaviour* of a network rather than the terms *dynamical behaviour* and *dynamics*.

**Modelling precedence** A second reading of the mathematical concept of time inherent to dynamical systems consists in understanding it as a simple *evolution parameter* defining no more than a relation of precedence between network configurations and without implying any notion of duration. If the two transitions  $x \longrightarrow y$  and  $x' \longrightarrow y'$  are both possible, then, with this point of view, it becomes coherent to accept that  $x \longrightarrow y$  may take much longer to happen than  $x' \longrightarrow y'$ , under certain circumstances, while, perhaps, under different circumstances, the opposite is true (*i.e.*,  $x' \longrightarrow y'$  takes longer than  $x \longrightarrow y$ ). Thus, different behaviours of the network can take place at different time scales although no additional precisions aim at distinguishing these time scales or the different possibilities they yield. The mathematical concept of time of dynamical systems is regarded as a *logical* version of time and it requires less information on the nature of transitions and on how they happen. Paths or trajectories simply are sequences of successive events. The time they take cannot be measured but the number of events or elementary events they involve can however be counted. Consequently, the questions that characterise this point of view on Boolean automata networks and dynamical systems are of the following form: *How many steps does the network take or is the network expected to take to reach a certain configuration or to start displaying a certain behaviour? What is (the most likely) network configuration that is reached after  $k$  steps? Can a given behaviour be observed after a certain other? What trajectories or behaviours are more likely?...*

**Modelling causes** One last point of view consists in ignoring altogether any associations that can be made between an intuitive idea of time (precedence as well as duration) and the theoretical features that follow from formalisation. This way, contrary to the case where Boolean automata networks are assimilated to dynamical systems, no more information than the transition graph is required. Any arbitrary Boolean automata network can be regarded as a *causal system* (*i.e.*, a state transition system) stripped from any notion of time. Obviously, there is no notion of duration associated to causal systems. But neither is there any meaningful notion of time precedence that follows naturally and non ambiguously from their definition. Indeed, when several transitions  $(x, y^k)$ ,  $k \in \mathbb{N}$ , are possible in the same network configuration  $x$ , then none of the network configurations  $y^k$  is the configuration that is reached after  $x$ . Each configuration  $y^k$  is only the result of one of several possible events which may occur with an unknown probability and within an unknown lapse of time. The notion of *moment* is therefore replaced by the notion of *possibility* and duration is replaced by a logical relation between causes and consequences. Two transitions  $(x, y)$  and  $(x, z)$  being possible means that  $x$  allows at least two different “*continuations*”,  $y$  and  $z$ . Causal systems are non-temporised systems in which the focus is placed on the degrees of freedom (*i.e.*, the set of possible state switches) that the network and the automata in it have in each configuration. Time-related questions such as those that have been mentioned in the previous paragraphs lose their immediate meaning. Further, although the problem of how to prune a transition graph in

order to make all trajectories deterministic can obviously be pertinent in the case of a dynamical system (provided additional information, such as a stochastic transition matrix  $P$  that specifies what transitions can indeed be ruled out), it is not in the more general context of state transition systems. Indeed, in this context, transitions are associated to no more information than that of their own existence. Thus, no transition of the system that is *a priori* possible can be disregarded *a posteriori*, even if it represents a highly improbable event. The only pertinent questions in the context of state transition systems are “existence questions” such as: *Can a configuration that satisfies a certain set of properties be reached from a given configuration  $x \in \mathbb{B}^n$ ?, Is a given behaviour possible?, Can this transition be made?, What new behaviours can be reached or become possible if some new transitions are added?...* One example of a problem that fits exactly into this framework is studied in [54]. The main question it addresses is whether the existence of synchronous transitions in a transition graph increases or decreases the possibilities in the network behaviour.

### 3.3 Example: modelling time and genetic regulation networks

In [15, 60, 62], the authors develop a formalism based on Boolean automata networks to model genetic regulation networks. This formalism integrates a formal notion of delay to account for the time flow that the regulations are subjected to in reality. It also assumes the two following statements:

**Hypothesis 1** *Only asynchronous transitions are possible.*

**Hypothesis 2** *All effective asynchronous transitions that update and activate (resp. deactivate) the same automaton take the same amount of time.*

More precisely, each automaton  $i \in V$  is assigned two values  $d_i^+ \in \mathbb{R}$  and  $d_i^- \in \mathbb{R}$ , called respectively its *activation* and *deactivation delays*. They represent the time it takes for automaton  $i$  to be updated and to change states:  $d_i^+$  (resp.  $d_i^-$ ) corresponds to the time  $i$  takes to switch from 0 to 1 (resp. from 1 to 0). By Hypothesis 2, delays do not depend on the context: whatever the network configuration  $x \in \mathbb{B}^n$  in which  $i$  is unstable ( $i \in \mathcal{U}(x)$ ), transition  $x \xrightarrow{-i} \bar{x}^i$  lasts  $t_i$  time units where  $t_i = d_i^+$  if  $x_i = 0$  and  $t_i = d_i^-$  if  $x_i = 1$ . By Hypothesis 1, this covers all possible transitions so, in the rest of this section, we use the following notations:

$$x \xrightarrow{-d_i^+} \bar{x}^i \text{ if } x_i = 0 \quad \text{and} \quad x \xrightarrow{-d_i^-} \bar{x}^i \text{ if } x_i = 1.$$

*Example 9.* Consider a network of size 2 whose structure is pictured by Figure 1 and whose set of local transition functions is:

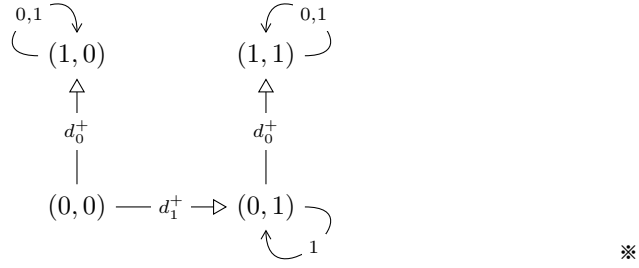
$$f_0 : \begin{cases} \mathbb{B}^2 \rightarrow \mathbb{B} \\ x \mapsto 1 \end{cases} \quad \text{and} \quad f_1 : \begin{cases} \mathbb{B}^2 \rightarrow \mathbb{B} \\ x \mapsto \neg x_0 \vee x_1 \end{cases}.$$





**Fig. 1.** Interaction graph of the network considered in Example 9.

The  $\text{ATG}^{\text{eff}}$  of this network, completed with the delay specifications, is given below:



An appreciable consequence of the addition of delays, in a modelling context, and one of their main purposes is to allow a natural deterministic interpretation of a network behaviour. Indeed, when two transitions are possible in a same configuration, we may suppose that the one that does effectively take place is the one that happens faster. Thus, in configuration  $(0, 0)$  of the network of Example 9, for instance, if  $d_0^+ < d_1^+$ , then transition  $(0, 0) \xrightarrow{d_0^+} (0, 1)$  can be supposed to be the one to occur. Otherwise, if  $d_0^+ > d_1^+$ , transition  $(0, 0) \xrightarrow{d_1^+} (0, 1)$  can be considered as the most likely.

**Interpreting asynchronicity** Let us analyse the meaning of Hypothesis 1 with respect to the formalism of Boolean automata networks. To do so, let  $x \in \mathbb{B}^n$  be a network configuration in which several automata are unstable. In particular, let  $i, j \in \mathcal{U}(x)$ . By definition, both automata  $i$  and  $j$  are “on the verge of” changing states in  $x$  but by Hypothesis 1, they cannot *both* change states in that configuration. In other terms, transitions

$$x \xrightarrow{i} \bar{x}^i \quad \text{and} \quad x \xrightarrow{j} \bar{x}^j$$

are both possible but transition

$$x \xrightarrow{\{i,j\}} \bar{x}^{\{i,j\}}$$

is not. This apparent conflict between the predefined theory of Boolean automata networks (which specifies that both automata can change states and have no reason not to) and Hypothesis 1 (which disallows them from changing states under

some circumstances) can be addressed by additional notions of time and duration. The delays implied by Hypothesis 2 and introduced in [15, 60, 62] provide these notions precisely. Indeed, labelling transitions by delays augments their meaning beyond that of a simple relation of precedence or causality as discussed in Section 3.2. And with this additional meaning, rather than stating that  $i$  and  $j$  cannot both change states, Hypothesis 1 can be understood as imposing that  $i$  and  $j$  cannot both change states *simultaneously*. It becomes significant and coherent to consider what happens *during* a transition  $x \longrightarrow y$ , between the moment where the network is in configuration  $x$  and the moment it reaches configuration  $y$ . This way, more precisely still, Hypothesis 1 can be taken as the claim that no two automata can *finish* changing states simultaneously.

With this interpretation of Hypothesis 1, let us suppose that in configuration  $x$ ,  $i$  and  $j$  are respectively subjected to the delays  $d_i \in \{d_i^-, d_i^+\}$  and  $d_j \in \{d_j^-, d_j^+\}$  and that:

$$d_i < d_j.$$

In configuration  $x$ , say at time 0, both automata start changing states since, according to the theory, they can. However at time  $d_i$ , automaton  $i$  has effectively changed states whereas automaton  $j$  has not yet had the time to. At this moment, only two situations are *a priori* coherent with both the theory of Boolean automata networks and our interpretation of Hypothesis 1:

1. Either  $j$  has become stable ( $j \in \overline{\mathcal{U}}(\overline{x}^i)$ ) in which case it must be that  $j$  is influenced (directly or indirectly) by  $i$  (*i.e.*, there is an arc or a path from  $i$  to  $j$  in the network interaction structure);
2. Or  $j$  is not influenced (neither indirectly nor directly) by  $i$  (and nor is its instability) and thus, the change of states of  $i$  (effective at the current time  $d_i$ ) has not affected  $j$  which can still change states in  $\overline{x}^i$  ( $j \in \mathcal{U}(\overline{x}^i)$ ).

The second case requires more attention and supplementary justifications. Indeed, in this case, Hypothesis 1 imposes that the trajectory:

$$x \xrightarrow{d_i} \overline{x}^i \xrightarrow{d_j} \overline{x}^{\{i,j\}}$$

is possible while the trajectory:

$$x \xrightarrow{d_i} \overline{x}^i \quad \overline{x}^{\{i,j\}}$$

is not. This means that even though  $j$  starts changing states in  $x$ , it must stop and start all over again in  $\overline{x}^i$ , making the whole process last  $d_i + d_j$  time units rather than  $\max\{d_i, d_j\} = d_j$ . Consequently, the change underwent by  $i$  in transition  $x \xrightarrow{d_i} \overline{x}^i$  which *a priori* involves  $i$  alone has an impact on the possibilities of  $j$ . And this impact is non-negligible since it concerns the duration of an event in a context where durations, precisely, have been given significance. There are two ways to preserve the consistency of the fact that  $i$  thus influences  $j$  in all cases and the formal basis of the theory of Boolean automata networks. One

can either suppose that two automata cannot be simultaneously unstable unless they have a mutual and symmetric influence on one another:

$$\forall i, j \in V, \exists x \in \mathbb{B}^n, \{i, j\} \in \mathcal{U}(x) \implies (i, j) \in A \text{ and } (j, i) \in A.$$

Or one can rule out simultaneity altogether and take on the following hypothesis:

**Hypothesis 3** *No two distinct events can start, finish or occur synchronously.*

Because the first solution is very restrictive, we choose the second and for the rest of this section, we accept Hypothesis 3. Let us, however, point out that this snag introduced by Hypothesis 1 highlights the difficulty, in a modelling context, of justifying restrictions or refinements brought to a theory. In particular, it shows that choosing to exclude some elementary transitions in the description of the behaviour of a Boolean automata network is a non-trivial choice when the aim is not simply theoretical convenience. Indeed, to defend this choice some strong arguments are needed (such as those required to justify Hypothesis 3 which is needed to support Hypothesis 1, as we have seen above) that can obviously not be drawn from the theory itself but must still remain coherent with it.

### Genetic regulation systems and Boolean automata networks with delays

To study how Boolean automata networks with delays model genetic regulation systems, let us now consider a simple genetic regulation system involving exactly  $n$  distinct genes  $G_0, \dots, G_{n-1}$ . Very schematically, each gene  $G_i$  may undergo a process by which the information it carries is “decoded” and “read” [64, 65] to induce the synthesis of one or several proteins. For each gene  $G_i$ , let  $P_i$  be one of these proteins products of  $G_i$ <sup>14</sup>. When  $G_i$  is being decoded and read, we say that it is *active*. The activation of genes  $G_i, \dots, G_{n-1}$  can be modelled by  $n$  Boolean variables  $g_0, \dots, g_{n-1}$  so that,  $\forall i \in V = \{0, \dots, n-1\}$ ,  $g_i = 1$  (resp.  $g_i = 0$ ) represents the activation (resp. the non-activation) of gene  $G_i$ .

Proteins can regulate the activation of genes. Thus, a gene  $G_i$ , via its protein  $P_i$ , may have a retro-action on the system of genes to which it belongs. For instance, it may influence the activation of gene  $G_j$ . If it does, this (indirect) regulation of  $G_j$  by  $G_i$  can be represented by an arc  $(i, j) \in A$  of an interaction graph  $\mathcal{G} = (V, A)$ . More precisely, and with respect to other protein influences that can act on  $G_j$ , it can be represented by a local transition function  $f_i : \mathbb{B}^n \rightarrow \mathbb{B}$ .

The effect of a regulation modelled by an arc  $(i, j) \in A$  is however subjected to there being enough proteins  $P_i$  in the cell. Here, for the sake of simplicity, we suppose that each protein  $P_i$  becomes able to regulate the activation of any other gene if and only if its concentration exceeds a certain threshold. If it does, the protein is said to be *active*. This hypothesis precisely allows a Boolean modelling of genetic systems: a protein can either be active with respect to all influences

<sup>14</sup> We consider only one protein for each gene for the sake of simplicity. Considering several would be more realistic and would not alter the basis and thrust of our argumentation. But it would considerably burden its formulation.

it may have on the system, or it may be inactive for all of them (if several levels of activation of  $P_i$  needed to be considered, it could not be modelled by just two states).

Let us say that gene  $G_i$  is *expressed* when protein  $P_i$  is active. And let us highlight the difference in our use of the two terms *active* and *expressed* concerning genes. Gene  $G_i$  may be active without being expressed. This usually happens right after the precise instant  $G_i$  is activated. Before it becomes expressed as well, that is, before the precise instant the concentration in  $P_i$  hits its activation threshold, a time lapse is required during which the concentration in  $P_i$  increases but remains under its activation threshold. This time lapse, precisely, is modelled by the delay  $d_i^+$ . Conversely,  $G_i$  may be inactive and expressed. Indeed, following the instant  $G_i$  is deactivated, it may remain expressed during some time. This time corresponds to the time it takes for the concentration in  $P_i$  to decrease as a result of the molecules degradation and of their not being renewed (since the synthesis of  $P_i$  is no longer “commanded” by  $G_i$ ), and fall below its activation threshold. It is modelled by the delay  $d_i^-$ . In summary, delays  $d_i^+$  and  $d_i^-$  both represent the time that  $P_i$  takes to respond accordingly to a “command” sent by  $G_i$ .

More formally, let us introduce  $n$  new Boolean variables,  $x_0, \dots, x_{n-1}$ , to model the expression of each gene  $G_i$ , or equivalently, to model the activation of each protein  $P_i$ :  $\forall i \in V$ ,  $x_i = 1$  (resp.  $x_i = 0$ ) models the expression of  $G_i$  (resp. its non-expression) and the activation of protein  $P_i$  (resp. its non-activation). Then, to model the regulation system comprised of genes  $G_i$  and their products  $P_i$ , let us consider a Boolean automata network of size  $n$ , whose interaction structure and local transition functions are as suggested above and whose automata states are the variables  $x_i$ ,  $i \in V$ . Thus, a network automaton  $i \in V$  rather represents the protein  $P_i$  than its coding gene  $G_i$ <sup>15</sup>. Let us consider a situation in which the activations of genes  $G_i$  are modelled by the vector  $g = (g_0, \dots, g_{n-1}) \in \mathbb{B}^n$  and the activations of the proteins  $P_i$  are represented by the network configuration  $x = (x_0, \dots, x_{n-1}) \in \mathbb{B}^n$ . To model such a situation, we use the following matrix which, abusing language, we call the *network configuration*:

$$\begin{bmatrix} x \\ g \end{bmatrix} = \begin{bmatrix} x_0 & \dots & x_{n-1} \\ g_0 & \dots & g_{n-1} \end{bmatrix}. \quad (14)$$

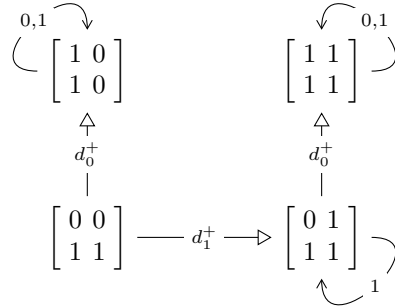
In [15, 60, 62], only configurations where  $g_i = f_i(x)$  are considered, *i.e.*, Hypothesis 4 figuring below is made. As a consequence,  $g_i = x_i$  is equivalent to automaton  $i$  being stable ( $i \in \bar{U}(x)$ ) and,  $\forall i \in V$ ,  $g_i = x_i$  is equivalent to the configuration being a stable configuration.

**Hypothesis 4** *The only possible network configurations have the following form:*

$$\begin{bmatrix} x_0 & \dots & x_{n-1} \\ g_0 & \dots & g_{n-1} \end{bmatrix} = \begin{bmatrix} x_0 & \dots & x_{n-1} \\ f_0(x) & \dots & f_{n-1}(x) \end{bmatrix}.$$

<sup>15</sup> This is only a convention that we set and which follows naturally from our choices of notation. Other presentations that lead to automata corresponding to genes rather than to proteins are possible.

*Example 10.* With the notation introduced in Equation 14 and with Hypothesis 4, the transition graph of Example 9 still has only four configurations, those that satisfy  $g_0 = 1$  and  $g_1 = \neg x_0 \vee x_1$ :

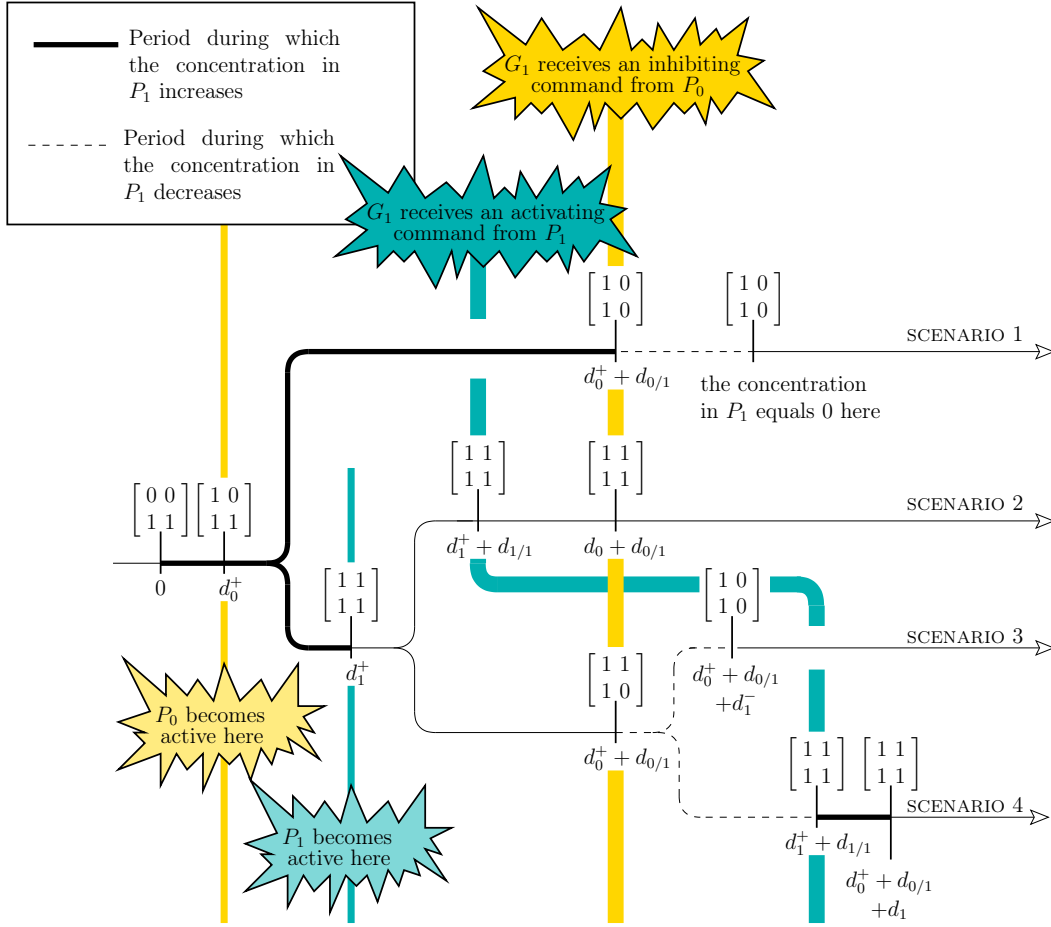


In particular, the following network configurations are supposed to be unrealisable:

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \quad *$$

Boolean automata networks with delays can thus serve as models in which a mathematical notion of time flow and duration is introduced to match the time that rules over the interactions of the real system they represent. As mentioned in the previous paragraphs, it is important to note, however, that the pertinence of the modelling strongly depends on the hypotheses that are added to the original theory. More specifically, it depends on the interpretations of these hypotheses and on their consistency with respect to prior interpretations of other features of the theory. We have already discussed above Hypothesis 1 and how its significance relies on the additional notion of time introduced by means of delays, and more precisely, how it relies on Hypothesis 3 (unless, as we have demonstrated, all network automata are supposed to interact two by two). We propose now to analyse further the role of hypotheses and the correspondence between this refined version of the theory of Boolean automata networks and the genetic regulation systems that they are intended to model.

**A rigorous adaptation and interpretation of Boolean automata networks with delays** By Hypothesis 3 (and *a fortiori* by Hypothesis 1), we have assumed that proteins do not receive commands from their coding genes *immediately* when these are emitted. For the precise same reason (*i.e.*, two distinct punctual events are necessarily separated in time by a lapse of time, however small it be), let us suppose similarly that genes cannot change states *simultaneously* to the changes of the states of their regulating proteins. Of course, one

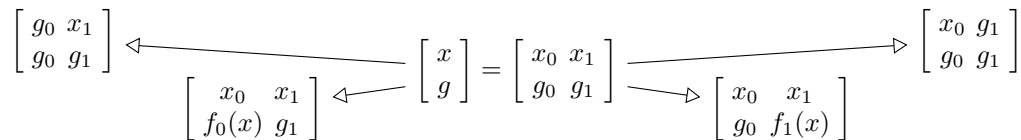


**Fig. 2.** Diagram representing possible behaviours of the regulation network of Examples 9 and 10 under the assumption that  $d_0^+ < d_1^-$ . Let us emphasise that no effort whatsoever has been put into representing the time scale. Consequently, despite the space that separates the representation of the dates  $d_0^+$  and  $d_0^+ + d_{0/1}$ , it is not excluded that  $d_0^+ \approx d_0^+ + d_{0/1}$ .

can argue that the two phenomena evoked are set in different time scales and that the second direction involves times lapses that are completely insignificant compared to those involved by the first. Nevertheless, the point is to respect Hypothesis 3 rigorously and rule out simultaneity altogether. Thus, for any couple of events that are considered, one event must necessarily occur *before* the other. Therefore, either the change of states of a gene and that of its regulating protein must be considered as one and only event, or the changes must be separated in time by a delay, possibly very small. We choose the second solution (by identifying gene state changes and protein state changes, the first solution demands

too severe renouncements of the original modelling). Thus, we denote by  $d_{i/j}$  the time it takes between a change of states of protein  $P_i$  and the according change of states of a gene  $G_j$  which is regulated by it. And we do not exclude the possibility that the new delays  $d_{i/j}$  be evidently negligible in comparison to the first delays introduced,  $d_i^+$  and  $d_i^-$ .

Let us concentrate on the Boolean automata network of Examples 9 and 10 and let us consider this network as a model of a genetic system with two genes  $G_0$  and  $G_1$  and their two protein products  $P_0$  and  $P_1$ . Under the three original hypotheses 1, 2 and 4, the behaviour of this network is given by the transition graphs of Examples 9 and 10. Under the additional but necessary Hypothesis 3, no network configurations (as defined by Equation 14) are excluded any longer. This implies in particular that Hypothesis 3 (and *a fortiori* Hypothesis 1) and Hypothesis 4 cannot strictly be satisfied at once. And since transitions still are asynchronous, in theory, in our example network of size two, four elementary transitions must be considered in every configuration (some of which may be null):



Rather than giving the complete new version of the transition graph of this network (which contains 16 configurations), let us suppose that  $d_0^+ < d_1^+$  and let us consider the diagram pictured in Figure 2. It describes informally four alternate scenarios that can occur under the current general Hypotheses 1, 2 and 3. All four of these scenarios start with the activation of  $P_1$  in the network configuration modelling the situation in which both genes are active but unexpressed. The system is supposed to be isolated and free of any exterior perturbations so that failures in protein concentrations (resp. in the expression of genes) due to other factors than the deactivation of their coding genes (resp. the inhibition of their regulating proteins) are ignored. In particular, cases in which  $g_0 = 0$  are disregarded. Let us comment on this new modelling of the system behaviour. First, several “kinds” of configurations

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

are involved. There are those in which the signal emitted by protein  $P_1$  addressed to its coding gene  $G_1$  has not (yet) been received and there are those in which it has. Figure 2 suggests that in the latter configurations, the activation of  $G_1$  is more stable than in the former in which the activation of  $G_1$  is re-enforced by the presence of an activating protein (rather than just being made possible by the absence of an inhibiting one). Second, besides their respective complexity and the transient trajectories they involve, Figure 2 and the transition graph of Example 10 yield similar results if one focuses on limit behaviours. These remarks may encourage one in arguing that Figure 2 represents a futile com-

plexification of the original modelling. However, let us recall that, importantly, this new modelling follows from a rigorous adaptation of the original theory of Boolean automata networks with delays and that it was made necessary to make the theory fit coherently to its priorly intended interpretation. Moreover, in this context where Hypotheses 1, 2 and 3 all need to be respected, the possible duality in the meaning of network configurations that are formally considered as one unique configuration, as well as the existence of different intermediary steps leading to the limit stable configurations are important. Both these features of the modelling cannot be disregarded *a posteriori* since they result directly and essentially from the work of theorisation and interpretation that was made *a priori*. In particular, if the network was actually a sub-network of a larger network, then some other automata may be affected by the states of the two automata that are considered. As a consequence, the behaviour of the whole network could be considerably affected by slight deviations from what is predicted by the original transition graph. These deviations could, for instance, consist in gene  $G_1$  being less stable in some configurations where  $x_0 = x_1 = g_0 = g_1$  than in others. Or they could be the intermediary steps through which the network may pass even if only to remain a fleeting instant. Since our new formalism is a rigorous refinement and adaptation of the original, the pertinence of the modelling it produces cannot be questioned unless that of the original version is. It can, however, be criticised for its increased complexity all the more so that Figure 2 proposes an incomplete description of the network behaviour. Indeed, in Scenario 3, not all signals emitted are received. Protein  $P_1$  is active during a certain period, *i.e.*, its concentration in the cell exceeds the threshold for some time and yet, its coding gene  $G_1$  ignores it. For the sake of formal rigour and consistency, the description should be completed with a sequel to this branch which would increase further the complexity of the description. Further, to interpret this new formal description and to gauge the feasibility and plausibility of any of the scenarios it mentions, relationships of precedence need to be established between the events that are considered. Consequently, delays (and perhaps protein concentrations) need to be compared precisely. But this leads to the conclusion that to exploit the new formalism, it must be refined further still, suggesting in the end that a continuous framework would be better suited for the modelling problems at hand here. Thus, an eventual change of modelling paradigm seems to be required.

Boolean automata networks with delays thus provide indeed a modelling of some real phenomena that takes time flow into account. The supplementary parameters that are integrated in these models apparently provide an enhancement of the original model that potentially allows to better understand the conditions required for one scenario to occur rather than another according to the delays that separate the events involved in each. But the modelling power of this formalism relies substantially on the non trivial interpretation of some formal hypotheses, in particular, the absolute non-simultaneity of any (lasting) couple of events and the assumption that whatever the current state of the cell, a protein always takes the same amount of time to change states.



## 4 Effective modelling and observing of a system

Once a formal well-bounded framework has been accepted and coherently associated to a correspondence with reality, the observing and modelling of the behaviour of a particular system can be carried out. The present section is set in this general context. More specifically, it lies on the hypothesis that neither the machinery of a system nor its movements can ever be effectively observed. Instead, it is the successive punctual positions or states of the system that can be. As an illustration, on the 9<sup>th</sup> of august 2011 at local time 12 : 30 : 56 in the place with geographic coordinates (45.662587, 1.789201), we may observe that proteins  $P_1$  and  $P_2$  are both present in very high concentrations in the liver cells of the rat Bobby. Two minutes later, we may note that the concentration of  $P_1$  is almost null. Further, we may perform the same experience at different dates, times, places and with different rats, and make similar observations whereas a second experience in which  $P_2$  is initially absent always yields different observations: the concentrations of  $P_1$  that are recorded always are approximately identical during several hours. In both experiences, no dynamical phenomenons such as concentration changes are observed, only their results are. In addition, the influence of protein  $P_2$  on the concentration of protein  $P_1$  is not observed effectively although several situations are observed from which this influence (or perhaps a different conclusion) may be inferred. Now, primarily, the very nature of modelling makes it impossible for any model to even come close to reproducing or describing a real system faithfully: from the start, the theorisation process can only aim at designing a framework that hopefully will allow models to reproduce, describe or explain partially some properties of the system. Here, we add the hypothesis that the only available information concerning a real system is derived from necessarily partial observations of its behaviour. In summary, in our context, the effective modelling of a real system is supposed to start with a series of observations of its behaviour. Once a “reasonable” collection of observations has been established, it becomes possible to build a model of the system behaviour. To derive from it a model of the system structure and underlying mechanisms, however, is a tricky task that necessarily encounters sources of incompleteness which call for supplementary approximations and hypotheses. And these new approximations and hypotheses have notable significance with regards to the reality/theory correspondence.

### 4.1 From observations of a system to a model of its underlying interactions

The very first step in the process of effective modelling consists in determining a set of formal configurations to model the states in which the system is observed. Here, we focus on systems that contain a finite number of interacting elements which are intended to be modelled by Boolean automata networks so the formal configurations that are chosen are Boolean vectors  $x \in \mathbb{B}^n$  where each Boolean coefficient  $x_i \in \mathbb{B}$  of  $x$  is supposed to represent the state of a system element identified by the label  $i < n$ . The choice of the set of configurations is crucial

because it requires a knowledge of what is the “interior” of the system that is modelled and what is its “exterior”. Indeed, to specify the size of the model network and decide what is the dimension  $n$  of the Boolean vectors, the number of elements interacting in the system needs to be known. This often means that the set of elements itself needs to be known. Therefore, in most cases, from the very start, modelling requires to assume or to know that the content and the frontiers of the system to be modelled are exactly those that are suggested by or derived from observations of it. In other terms, unless certainty may be established on this subject, the following hypothesis is needed:

**Hypothesis 5** *The number  $n \in \mathbb{N}$  of interacting automata in the network is known.*

Then, the *movements* or changes of the system can be considered for modelling. Let us suppose that  $x \in \mathbb{B}^n$  is a configuration that models the system state which has been observed right before the system state modelled by configuration  $y \in \mathbb{B}^n$ . Then, the change underwent by the system between these two observed states is modelled by the couple  $(x, y) \in \mathbb{B}^n \times \mathbb{B}^n$ . Such couples are called *observed transitions* and denoted as follows:

$$x \rightsquigarrow y.$$

It is important to note that observed transitions are not necessarily network transitions. This will become clear in the sequel. The set of all observed transitions derived from observations of a system defines an *observed transition graph* whose role is precisely to model the system behaviour.

With an observed transition graph  $\mathcal{T}_{obs}$  in hand, the modelling can be pushed a step further. The aim is now to infer from  $\mathcal{T}_{obs}$  knowledge concerning the underlying mechanisms that contribute in producing the events that are described in  $\mathcal{T}_{obs}$ . In short, this third step of modelling aims at determining the causes of the effects observed. Thus, the information carried by  $\mathcal{T}_{obs}$  must be analysed and exploited in order to define a set of local transition functions  $\mathcal{F}$  and a network structure  $\mathcal{G}$  which must both be coherent with  $\mathcal{T}_{obs}$ , with the hypotheses that were made beforehand, with the underlying theory, and with its correspondence with reality, pre-defined by the prior work of theorisation. A natural way to do this is to derive  $\mathcal{F}$  using one of Equations 11, 12 and 13 or to exploit Algorithm 1 and then derive  $\mathcal{G}$  from  $\mathcal{F}$ . However, as mentioned in Section 2.10, this method requires some additional information and failing which, some additional hypothesis concerning the *nature* of the transitions in  $\mathcal{T}_{obs}$ . As a consequence, if  $\mathcal{F}$  and  $\mathcal{G}$  are defined this way and if they are effectively intended to be used, studied or exploited to derive additional information on the original system, then the following question needs to be answered: how can  $\mathcal{F}$  and  $\mathcal{G}$  be interpreted, *i.e.*, what modelling power can they be granted? In the absence of a complete and satisfiable answer to this question some new hypotheses need to be made.

The rest of this section proposes to illustrate some of these hypotheses and discuss their meaning and impact. In the examples that are taken, for the sake

of their simplicity, Boolean automata networks serve as models of Boolean automata networks. In other terms, the networks play both the parts of observed system to be modelled and model of the observed system. One might imagine that a computer program simulates the behaviour of a Boolean automata network  $N$  and prints out on a monitor the states it takes. Of course, when the program is executed, on the one hand, the program may be designed to hide some parts of the information and on the other hand, our diligence in watching the screen may not be perfect. We may, for instance, only look periodically at it, or from time to time at random, or we may perhaps forget to look at it altogether. In any case, the lists of successive configurations that are observed yield a transition graph  $\mathcal{T}_{obs}$  from which, as mentioned above, sets of automata, of interactions and of local transition functions may be inferred to define a new Boolean automata network  $N'$  supposed to model the original network  $N$ .

*Example 11.* Let us first illustrate the importance of Hypothesis 5. To do so, let us suppose that the observed transition graph  $\mathcal{T}_{obs}$  contains only the two following transitions:

$$(1, 0) \rightsquigarrow (1, 1) \quad \text{and} \quad (0, 0) \rightsquigarrow (0, 1).$$

By Hypothesis 5, we can start by deriving that the set of configurations of the observed network  $N$  is  $\mathbb{B}^2$  and that it involves two automata which are referred to here as automaton 0 and automaton 1. Choosing to model these two automata by two automata with the same names and using Equations 13 or 12 (*i.e.*, assuming transitions of  $\mathcal{T}_{obs}$  are elementary) yields the following local transition functions:

$$f'_0 : x \in \mathbb{B}^2 \mapsto x_0 \quad \text{and} \quad f'_1 : x \in \mathbb{B}^2 \mapsto 1.$$

The resulting model  $N'$  of  $N$  has the same behaviour as  $N$  (its set of possible transitions is exactly those of  $\mathcal{T}_{obs}$ ). It has  $\mathcal{F} = \{f'_0, f'_1\}$  as set of local transition functions and the digraph pictured below as interaction structure:



If Hypothesis 5 were not satisfied, however,  $N$  could contain a third automaton, automaton 2, that observations of  $N$  give no evidence of. Then, when transition  $(1, 0) \rightsquigarrow (1, 1)$  is observed, it could be that what is really occurring is a series of transitions that has the same effect of increasing the state of automaton 1 but involve the hidden automaton 2:

$$(1, 0, 0) \longrightarrow (1, 0, 1) \longrightarrow (1, 1, 1),$$

or perhaps just:

$$(1, 0, 1) \longrightarrow (1, 1, 1).$$

In that case, according to whether automaton 0 activates automaton 2 or not, or according to whether  $(1, 0, 0) \longrightarrow (1, 0, 1)$  is possible or not, the local transition function of automaton 2 could equal:

$$f_2 : x \in \mathbb{B}^3 \mapsto x_0 \quad \text{or} \quad f_2 : x \in \mathbb{B}^3 \mapsto 0.$$

That of automata 0 and 1 could respectively equal:

$$f_0 : x \in \mathbb{B}^3 \mapsto x_0 \quad \text{and} \quad f_1 : x \in \mathbb{B}^3 \mapsto x_1 \vee x_2.$$

As a consequence, rather than the interaction graph of  $N'$  that suggests that automata 0 and 1 are independent and depend only on themselves, network  $N$  could instead have one of the interaction graphs pictured in Figure 3.  $\ast$



**Fig. 3.** Possible interaction graphs of Boolean automata network whose transition graph  $\mathcal{T}_{obs}$  presented in Example 11.

Example 11 shows that relying on Hypothesis 5 in the construction of a model  $N'$  of a system  $N$  may possibly be responsible for the inferring of a structure that differs significantly from the reality of  $N$ . This way, Hypothesis 5 may cause to mistaken the cause of an event that is observed (the change of state of automaton 1 in Example 11, for instance). Now, let us consider some new hypotheses that apply more specifically to the transitions of  $\mathcal{T}_{obs}$ .

First, Hypothesis 6 below allows to focus effectively on  $\mathcal{T}_{obs}$  and on any information (such as  $\mathcal{G}$  and  $\mathcal{F}$ ) that can be drawn solely from it as suggested above. Indeed, it assumes that no exterior force or reason is responsible even partially for an observation that is made:

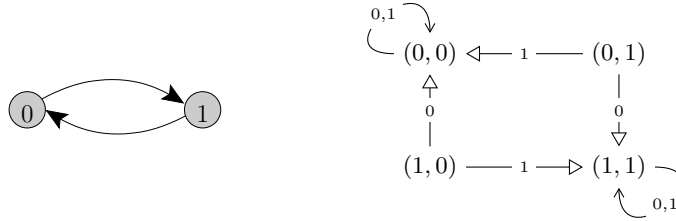
**Hypothesis 6** *Every behaviour of the system is caused by factors that are contained in its own definition.*

*Example 12.* Let us suppose that the observed Boolean automata network  $N$  satisfies the following properties. It contains two automata, automaton 0 and automaton 1 whose local transition functions are defined by:

$$\forall x \in \mathbb{B}^2, f_0(x) = x_1 \quad \text{and} \quad f_1(x) = x_0.$$

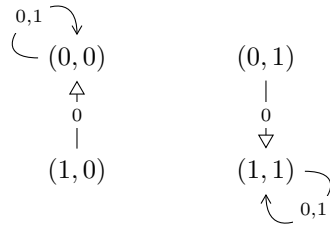
Its interaction graph is the digraph pictured on the left of Figure 4 and, for some reason,  $N$  can only perform asynchronous transitions and it can potentially perform them all. Its behaviour, in the absence of any exterior perturbations is described by the ATG<sup>eff</sup> on the right of Figure 4.

Now, let us suppose that  $N$  is subjected to some exterior forces that impose restrictions on its behaviour and that these restrictions can be translated in terms of updating constraints. For instance, it could be that active automata tend to be updated before inactive automata. In that case, transitions  $(0, 1) \xrightarrow{0} (1, 1)$



**Fig. 4.** Interaction graph and  $\text{ATG}^{\text{eff}}$  of the Boolean automata network of Example 12.

and  $(1,0) \xrightarrow{1} (1,1)$  in the  $\text{ATG}^{\text{eff}}$  of Figure 4 may be considered as unlikely or impossible. Here, let us suppose differently that an exterior force causes automaton 0 to be much faster in switching states than automaton 1 is. Thus,  $N$  behaves as if it were obeying to the sequential update schedule  $\delta \equiv \{0\}\{1\}$ . An outside observer that is aware that  $N$  only performs asynchronous transitions might possibly ignore that  $N$  is submitted to any updating constraints and as a consequence might record the following transition graph  $\mathcal{T}_{\text{obs}}$ :



If this were the case, the model  $N'$  of  $N$  would have local transition functions defined by:

$$\forall x \in \mathbb{B}^2, f'_0(x) = f'_1(x) = x_1$$

and the following structure:



The influence that automaton 0 has on automaton 1 would thus not be revealed at all because of the precedence of 0-updates over 1-updates. \*

The example above highlights the non-triviality of choosing to ignore the possibility that the network behaviour might be subjected to exterior influences that cannot be formalised as proper automata influences. In this example, focus on the particular but important case of influences that can be taken into account through the defining of an update schedule.

The next hypothesis specifies that every system behaviour that should have been observed, with respect to the “observation protocol”, has indeed been observed and modelled in  $\mathcal{T}_{obs}$ .

**Hypothesis 7** *With respect to the nature of observed transitions,  $\mathcal{T}_{obs}$  contains all paths and behaviours of the system model.*

Thus, if the system is supposed to have been observed often enough so as to witness each of its least changes then, by Hypothesis 7,  $\mathcal{T}_{obs}$  is the GTG or the  $\text{GTG}^{\text{eff}}$  of its model. If, on the contrary, the system is not supposed to have been observed often enough to acknowledge every elementary transition, then, Hypothesis 7 imposes that  $\mathcal{T}_{obs}$  still contain a “representation” of any possible system behaviour. If transition  $x \longrightarrow y$  is a (possibly elementary) transition that models a system change of states, then, by Hypothesis 7, it must be taken into account: the path that involves this transition must be represented in  $\mathcal{T}_{obs}$ , even if not elementarily. In some sense, Hypothesis 7 is a stronger version of Hypothesis 6 that assumes that all behaviours that an observer has no knowledge of are impossible under all circumstances. Conversely, it means that any change that is not observed is not possible, *i.e.*, any transition that does not belong to  $\mathcal{T}_{obs}$  either is contained in a path of  $\mathcal{T}_{obs}$  or is impossible altogether. As a consequence, Hypothesis 7 implies Hypothesis 8 below which relates (if not equates) the exercise that consists in observing movement to that of observing fixity.

**Hypothesis 8** *A configuration with out-degree null in  $\mathcal{T}_{obs}$  models a stable configuration of the system.*

*Example 13.* Let us suppose that only one transition of the Boolean automata network  $N$  has been observed:

$$(1, 0) \rightsquigarrow (1, 1).$$

Using Equation 12, we derive a model network  $N'$  of size 2 with the following local transition functions:

$$f'_0 : \begin{cases} \mathbb{B}^2 \rightarrow \mathbb{B} \\ x \mapsto x_0 \end{cases} \quad \text{and} \quad f'_1 : \begin{cases} \mathbb{B}^2 \rightarrow \mathbb{B} \\ x \mapsto x_0 \vee x_1 \end{cases}.$$

The interaction structure of  $N'$  is:



Supposing that Hypothesis 7 is actually not satisfied because, for instance, transition  $(0, 1) \longrightarrow (1, 1)$  has been missed, it could be that the local transition function of automaton 0 in  $N$  is in reality  $f_0 : x \mapsto x_0 \vee x_1$  rather than function  $f'_0$ . And the structure of  $N$  rather looks like:



than like the structure of the model  $N'$ . Missing some observations leads here to miss some interactions. Conversely, missing some observations can lead to infer interactions that do not exist. Indeed, this happens if network  $N$  can perform transition  $(0, 0) \longrightarrow (0, 1)$  although it has not been observed. If this is the case, the local transition function of automaton 1 in  $N$  does not equal  $f'_1$  either but, instead, the constant function  $f_1 : x \mapsto 1$  and the structure of  $N$  is as follows:



Thus, in reality, automaton 1 of  $N$  could be submitted to no influence from automaton 0 contrary to what is suggested by model  $N'$ . Further, combining this latter example to Example 11, we find that it could also be that  $N$  has a structure that looks like that of Figure 3 *a*. (this could happen if automaton 2 was not observed and transition  $(0, 0) \longrightarrow (0, 1)$  was missed).  $\ast$

The examples above highlight how failing to observe some of the possible movements of the system may hinder the correct reconstruction of its set of interactions by either leading to the inference of non-existing interactions or missing some existing ones.

In some cases, to exploit  $\mathcal{T}_{obs}$ , one may hypothesise on the granularity of events as in the last two hypotheses we mention here.

**Hypothesis 9**  $\mathcal{T}_{obs}$  is an elementary transition graph.

*Example 14.* Let  $\mathcal{T}_{obs}$  be the following digraph:



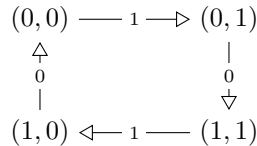
Assuming all transitions in  $\mathcal{T}_{obs}$  are elementary and using Equation 13, a model  $N'$  may be derived with the following local transition functions:

$$f'_0 : x \in \mathbb{B}^2 \mapsto \neg x_0 \quad \text{and} \quad f'_1 : x \in \mathbb{B}^2 \mapsto \neg x_1.$$

and the structure pictured below:



According to this model, the two automata of  $N$  do not interact. However, let us suppose that  $\mathcal{T}_{obs}$  has been derived by observing the configuration of  $N$  only once per time unit, while, in reality, the network  $N$  performs two transitions per time unit. Contrary to Hypothesis 9,  $\mathcal{T}_{obs}$  is not elementary. Further, let us suppose that the synchronous transitions that are observed in  $\mathcal{T}_{obs}$  are actually due to a series of two asynchronous transitions so that the ATG<sup>eff</sup> of  $N$  is:



and its structure is the digraph below:



Example 14 shows that assuming that observed transitions are elementary is not a negligible choice either, in terms of modelling. In the same lines, the next example shows that this is true even when transitions seem to be asynchronous.

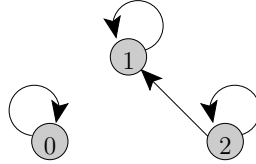
*Example 15.* Let  $N$  be of size 3 and let us suppose that its behaviour has been observed starting in each of its 8 different configurations. As a result of this experimentation the following list of possible transitions has been obtained:

$$\begin{aligned} (0, 1, 0) &\rightsquigarrow (0, 0, 0) & (1, 1, 0) &\rightsquigarrow (1, 0, 0) \\ (1, 0, 1) &\rightsquigarrow (1, 1, 1) & (0, 0, 1) &\rightsquigarrow (0, 1, 1) \end{aligned}$$

Assuming, by Hypothesis 9 that all four of these transitions are elementary (and asynchronous), a model  $N'$  can be derived with the following local transition functions:

$$f'_0 : x \in \mathbb{B}^3 \mapsto x_0, \quad f'_1 : x \in \mathbb{B}^3 \mapsto x_1 \vee x_2 \quad \text{and} \quad f'_2 : x \in \mathbb{B}^3 \mapsto x_2.$$

and the following interaction graph:



This modelling of  $N$  by  $N'$  suggests that the only interaction existing between two different automata of the network consists in the activation of automaton 1 by automaton 2. Supposing again that Hypothesis 9 is not satisfied, it might be that  $N$  is actually updated with the update schedule  $\delta \equiv \{0, 1\}, \{1, 2\}, \{0, 2\}$  of period 3. In that case, the four apparently asynchronous transitions that are observed are actually rough approximations of the four sequences of transitions that figure below. As before, this loss of information may be explained for instance by the fact that only one observation per time unit was made while  $N$  performed 3 transitions per time unit.

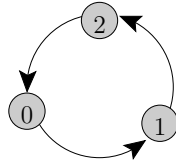
$$\begin{aligned} (0, 1, 0) &\xrightarrow{\{0,1\}} (0, 0, 0) & (1, 1, 0) &\xrightarrow{\{0,1\}} (0, 1, 0) \xrightarrow{\{1,2\}} (0, 0, 1) \xrightarrow{\{0,2\}} (1, 0, 0) \\ (1, 0, 1) &\xrightarrow{\{0,1\}} (1, 1, 1) & (0, 0, 1) &\xrightarrow{\{0,1\}} (1, 0, 1) \xrightarrow{\{1,2\}} (1, 1, 0) \xrightarrow{\{0,2\}} (0, 1, 1) \end{aligned}$$



It can be checked that in this case, the local transition functions of the three automata of  $N$ , rather than being equal to the functions  $f'_i$  of the model  $N'$ , could equal the following

$$\forall x \in \mathbb{B}^3, \quad f_0(x) = x_2, \quad f_1(x) = x_0, \quad f_2(x) = x_1,$$

so that the interaction structure of  $N$  would be a circuit:



※

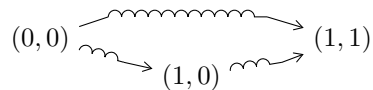
Thus, as demonstrated by Example 15, even an apparently asynchronous transition can in reality be a series of synchronous transitions. Let us note that generally, when  $\mathcal{T}_{obs}$  equals a transition graph of the form  $\mathcal{T}_\delta$  where  $\delta$  is a block-sequential update schedule, then, Equation 11 yields a model  $N'$  in which, informally, all the information of  $N$  has been “condensed” due to the application of  $\delta$  and to the transitivity of sequences of interactions [23, 33, 37]. As mentioned in Section 2.10, however, the condensed version of the information cannot be unravelled unless  $\delta$  is known precisely (see Algorithm 1).

Finally, when, contrary to the previous paragraph,  $\mathcal{T}_{obs}$  is not supposed to be elementary (for instance, because a specific update schedule is assumed) then, the information on the system carried by  $\mathcal{T}_{obs}$  is incomplete unless the series of system changes that occur between two successive observations of the system can be reconstructed.

**Hypothesis 10** *The additional information required to break down each non-elementary transition of  $\mathcal{T}_{obs}$  into an elementary path is known.*

Hypothesis 10 resembles Hypothesis 9 but is stronger. While Hypothesis 9 assumes that there is nothing more to each transition than what is observed, Hypothesis 10 assumes knowledge of what more there is to the transitions that are observed and considered non-elementary.

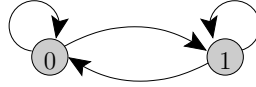
*Example 16.* Let  $\mathcal{T}_{obs}$  be the following observed transition graph:



where the bottom two transitions are supposed to be elementary and the remaining transition  $(0,0) \rightsquigarrow (1,1)$  is supposed to be an approximation of the elementary path  $(0,0) \rightsquigarrow (1,0) \rightsquigarrow (1,1)$ . Such hypotheses might for instance be made naturally if  $N$  is considered unable of synchronicity and if transition  $(0,1) \longrightarrow (1,1)$  is supposed to be impossible because it has not been observed. This, with Hypothesis 7 and Equation 12, yields a model  $N'$  which has the following local transition functions:

$$f'_0 : x \in \mathbb{B}^2 \mapsto x_0 \vee \neg x_1 \quad , \quad f'_1 : x \in \mathbb{B}^2 \mapsto x_0 \vee x_1$$

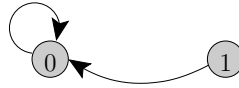
and the following interaction graph:



Suppose now that Hypothesis 10 is wrongly assumed and all transitions observed are not necessarily asynchronous.  $N$  is able to perform synchronous transitions and in particular, it can perform transition  $(0, 0) \xrightarrow{\{0,1\}} (1, 1)$  in one step. Then, it could be that the local transition function of automaton 1 is actually:

$$f_1 : x \mapsto 1$$

and its interaction structure is:



Thus, wrongly assuming Hypothesis 10 can also lead to deriving the existence of interactions that do not take place in reality. \*

To make up for the lack of information, modelling needs to put forward and rely on some supplementary hypotheses such as Hypotheses 5 to 10. This proves the importance of hypotheses. On the one hand, they allow modelling to happen. On the other hand, as we have endeavoured to show with the examples of this section, they can mislead it. In particular, they can cause the interaction structure of a system and of its model to differ significantly, mainly due to the fact that even in very simple instances of Boolean automata networks, there might be different possible causes to a same effect.

## 4.2 Defining networks

Relying on the previous sections, we now can propose the following formal definition of Boolean automata networks:

**Definition 1.** *A Boolean automata network  $N$  of size  $n \in \mathbb{N}$  is defined by a set of  $n$  local transition functions:*

$$N = \mathcal{F} = \{f_i \mid 0 \leq i < n\}.$$

Let us note that by Section 2.10, from a set  $\mathcal{F}$  of local transition functions, the GTG of the network defined by  $\mathcal{F}$ , can be computed. Conversely, given the GTG of the network,  $\mathcal{F}$  can also be inferred. As a consequence, a Boolean automata network is given equivalently to Definition 1 by its GTG.

**Definition 1 vs. the interaction structure** Let us first recall that with this definition, given a Boolean automata network  $N = \mathcal{F}$  of size  $n$ , the interaction structure of  $N$  can be inferred in time  $\mathcal{O}(2^n)$  (see Section 2.10). Now, comparing

with a possible definition of Boolean automata networks by their interaction structures, we find that, contrary to such a definition, Definition 1 implies a first implicit hypothesis:

**Hypothesis 11** *An automaton either obeys all of its influences at once (when it is updated), or it obeys none.*

In other words, by our choice of definition, we assume that any updated automaton  $i \in V$  can under no circumstances ignore some of its influences  $(j, i) \in A$ . However, let us note that this restriction can be partially resolved by a judicious design of the local transition functions  $f_i$ . For instance, if originally,  $f_i(x) = x_j$ , then  $i$  can be made independent of  $j$  when  $j$  is inactive by replacing  $f_i$  by  $x \mapsto x_j \vee x_i$ . This way of making  $i$  independent of  $j$  can nevertheless only be done relatively to a given network configuration or set of network configurations<sup>16</sup>. This leads us to the following remark. Definition 1 implies another non-trivial implicit hypothesis:

**Hypothesis 12** *The result of the interactions that take place in a given configuration depends only on that configuration.*

Thus, by Definition 1, we assume that given a network interaction structure  $\mathcal{G} = (V, A)$ , the nature of the influence represented by the arc  $(j, i) \in A$ , when this influence is effective, depends only on the current network configuration. Consider a network of size 2, for instance, where automaton 1 depends on automaton 0 and the local transition function affected to automaton 1 is  $f_1 : x \in \mathbb{B}^2 \mapsto x_0 \in \mathbb{B}$ . Then, in configuration  $x = (1, 0)$  whatever the current environment of the network, if the influence of automaton 0 on automaton 1 does take place (*i.e.*, if automaton 1 is updated), then automaton 0 causes automaton 1 to take state 1. Definition 1 and more specifically Hypothesis 12 disallow any situation in which the active automaton 0 deactivates automaton 1 in configuration  $x$ .

In spite of the disputable limitations imposed by Hypotheses 11 and 12, we still prefer Definition 1 to a less restrictive definition of Boolean automata networks by their interaction structures. The reason is mainly arbitrary but motivated, however, by the belief that the *nature* of each interaction that can potentially take place in a network (rather than just their existence) is an essential feature of a network.

**Definition 1 vs. the transition graph** Similarly we have chosen not to define a Boolean automata network by a transition graph (other than its GTG).

<sup>16</sup> More generally, let us suppose that  $f_i(x)$  is given in CNF. For any Boolean formula  $\phi(x)$  in CNF, let  $\phi(x)[0/x_j^*]$  equal the formula  $\phi(x)$  in which every occurrences of both the literal  $x_j$  and its negation  $\neg x_j$  are replaced by 0 (*e.g.*, if  $\phi(x) = (x_1 \vee \neg x_2 \vee \neg x_4) \wedge (x_2 \vee x_3)$ , then  $\phi(x)[0/x_2^*] = (x_1 \vee \neg x_4) \wedge x_3$ ). Then, automaton  $i$  can be made to ignore the influence  $(j, i) \in A$  in configuration  $y \in \mathbb{B}^n$  if  $f_i$  is replaced by the function  $x \mapsto f_i(x) \vee f_i(y)[0/x_j^*]$ .

In short, the reason for this is to avoid restricting a network to a system that has only one behaviour. Indeed, if, contrary to Definition 1, a Boolean automata network  $N$  is defined by a transition graph  $\mathcal{T}$  (which is not the GTG of  $N$ ), then,  $N$  can only behave according to  $\mathcal{T}$ . Of course, this definition may indeed appear better suited for a modelling context involving steps similar to those described in Section 4.1 (which start with the formalisation of an observed system behaviour in terms of a transition graph  $\mathcal{T}$ ). However, it is important to note again that in such a context, it is the *transitions* of  $\mathcal{T}$  rather than the interactions of  $N$  that are initially designed to model a portion of reality. All the examples of Section 4.1 show that this difference is significant. The pertinence of the modelling embodied by these transitions is contingent on the circumstances in which the real system is observed as well as on *how* it is observed. In addition, as developed in Section 4.1, the interactions that take place in the network  $N$  are only *inferred theoretically* based on some non-trivial hypotheses such as Hypotheses 5 to 10. The meaning, in terms of modelling, of the interactions derived this way is thus subjected to the relevance of the hypotheses that are made. However, identifying the observed transition graph with the model network  $N'$  itself fails at putting forward clearly these hypotheses which are instead relegated into a pre-accepted definition. As a consequence, Hypotheses 5 to 10, in particular, are made implicitly and systematically without being questioned.

Definition 1, on the contrary, either makes irrelevant Hypotheses 5 to 10, or it requires that they be made explicit. Indeed, since Definition 1 is equivalent to defining a network by its most general behaviour, *i.e.*, its GTG, any sub-graph of this transition graph can be considered as a possible elementary behaviour of the network. Further, the same is true for any other graph whose arcs can be broken down into paths that belong to the GTG. As a consequence, Boolean automata networks defined by Definition 1 may be considered as models of systems that possibly behave differently in different environments. Hypothesis 6, in particular, is made superfluous. More generally, given a network  $N = \mathcal{F}$  whose interactions are known *a priori* according to Definition 1, in any transition graph representing the behaviour of  $N$ , all transitions have non-ambiguous causes deriving directly from the network definition itself. Hypotheses 5 to 10 are irrelevant in this case. And in a converse situation similar to those discussed in Section 4.1 in which the interactions of a network are to be inferred from a partial description of its behaviour, Definition 1 makes Hypotheses 5 to 10 obviously needed so they cannot be ignored. The intrinsic “incompleteness” of Definition 1 (incompleteness in the sense that Definition 1 does not specify the network behaviour precisely) imposes that any supplementary theoretical argument that is used to give a ruling on the network behaviour be either justified or put forward as purely hypothetical and convenient.

## 5 Discussion

Modelling generally involves an intermediary step that has not been mentioned in this paper. This step, purely theoretical, is set between theorisation and ef-

fective modelling. It consists in manipulating objects of the theory alone and making series of logical inferences to prove or simulate new theoretical properties that will hopefully be interesting for the rest of the modelling process. Its main difficulty is to make proper use of the rules of classical logic commonly used in mathematics. Theorisation and effective modelling, on the contrary, are ubiquitous steps that require to go back and forth between reality and theory and that are concerned by any change brought to the theory or to the set of modelled characteristics of the real system. Generally, to compare reality with theory and to elaborate theory on the basis of reality makes indispensable a thorough and coherent definition and bounding of the modelling process itself. For this reason, theorisation and effective modelling involve a considerable difficulty of which the intermediary purely formal step is immune. This difficulty is to navigate safely but constructively between experiences and observations of reality on one side, and mathematical abstractions of it on the other.

Sections 3 and 4 have both emphasised how the safeguard of consistency throughout the modelling process is indeed an issue. In the respective contexts of theorisation and effective modelling, the intrinsic and dual incompleteness of any knowledge as well as of any representation of reality needs to be overcome. As we have endeavoured to highlight in this paper, hypothesising is a required but tricky solution to this. Consider Hypothesis 1, for instance. Section 3.3 argues that no pre-existing interpretation of the theory of Boolean automata networks can serve or yield a valid interpretation for this hypothesis. Generally, an explanation of a formal choice cannot be drawn solely from the theory itself (the theory cannot justify itself). But it cannot either rely exclusively on knowledge or observations of the real system or category of real systems being modelled. Indeed, arguments drawn from such knowledge and observations need to be consistent with the pre-defined reality/theory correspondence. Thus, to justify why, under Hypothesis 1, certain possible elementary transitions are ignored and more precisely why the network is supposed *not* to follow the “maximum speed gradient”, a new feature of reality must be put forward as argued in Section 3.3. First, a strong supplementary notion of time must be taken into account. Second, in addition to this, since the fortuitousness of two events finishing simultaneously is not enough (see Section 3.3) to justify Hypothesis 1 non-ambiguously, one must also consider an *absolute* notion of non-simultaneity that forbids any two events to overlap in time. The significance of this can be seen in particular by noticing that the modelling meaning of the local interaction graphs  $\mathcal{G}(x)$ ,  $x \in \mathbb{B}^n$ , is quite limited under these assumptions that are intended to support Hypothesis 1. Indeed, except for their existing at the same instant, the interactions in these digraphs are otherwise completely independent. And this absolute non-simultaneity must then be implanted rigorously into the pre-defined theory and be connected consistently with the pre-existing reality/theory correspondence. The refinement represented by Hypothesis 1 thus obviously calls for another refinement which is the time notion brought by delays. But, as the end of Section 3.3 shows, the consistency requirement also calls for further refinements. And, in turn, so does the need for non-ambiguous and plausible predictions.

And each refinement adds complexity to the modelling which increases the difficulty in the interpretation of the theory. This looming necessity for series of levellings of the approximations that are made by theorisation runs the risk of ultimately aiming towards a different modelling paradigm (in the case analysed in Section 3.3, the series seems to aim towards a differential formalism) which would necessarily disavow the present modelling paradigm by disavowing some of its founding hypotheses (a continuous formalism obviously denies the founding hypotheses of a discrete modelling such as a Boolean modelling). Thus, model refining, *i.e.*, the augmenting of the modelling and interpretation maps associated to a theory (see Page 18) as well as augmenting the theory itself is limited on one hand by the necessity to maintain consistency, and on the other hand by the risk of increasing the complexity of the modelling and its interpretation to an extent where a different model is required. The existence of these limits favours the two following ideas.

First, the ins and outs of a model (the theory and the reality/theory correspondence) must be understood and fathomed so that no unnecessary refinement be added. Thus, in a context where duration is not intended to be modelled specifically, one could exploit the original definition of transitions of Boolean automata networks to model a “duration-free” version of delays, bypassing the thorny problem of simultaneity. Indeed, all elementary transitions that are possible in an arbitrary configuration  $x \in \mathbb{B}^n$  (*i.e.*, all arcs outgoing  $x$  in the GTG) may simply be seen as representing the possibility that in  $x$  one or several automata may end up ulteriorly having changed states. If  $i, j \in \mathcal{U}(x)$  are two automata that can both change states in  $x$ , then the synchronous elementary transition  $x \xrightarrow{\{i,j\}} \bar{x}^{\{i,j\}}$  does not necessarily mean that the change of states of  $i$  and the change of states of  $j$  have *finished* simultaneously. Actually, it cannot mean this if time is not modelled (the notion of “simultaneity”, in particular, is obviously based on that of time). With the most general and basic interpretation of the theory of Boolean automata networks, synchronous transitions can only be interpreted, “at the best”, as models of changes that have been produced during overlapping periods of time: a transition  $x \xrightarrow{\{i,j\}} \bar{x}^{\{i,j\}}$  can be taken to mean that neither  $\bar{x}^i$  nor  $\bar{x}^j$  are visited long enough by the network (if visited at all) for there to occur, in these configurations, a decisive event that could divert the network from its path from  $x$  to  $\bar{x}^{\{i,j\}}$ .

The second idea that derives from the evidence of the limits of refining a model concerns the intrinsic incompleteness of modelling and its benefit. Levelling the incompleteness of a model to make it apparently better fit reality can make it hard to bound properly the properties that are supposed to be modelled and embrace the approximations and hypotheses that were made in order to do so. Yet, modelling draws its pertinence precisely from its incompleteness. It is based on approximations of reality that can be very rough, perhaps even incorrect, and on the formal hypotheses to which they translate (such as the discreteness of events and time). Whether these hypotheses be supported by reasonable arguments or not does not necessarily impact directly on the pertinence of a model. Indeed, one of the main purposes of hypotheses is to be questioned and eventu-

ally discarded. Convincing model predictions do not produce an explanation of reality in themselves. An analysis of the hypotheses on which is based the model, however, may. On one hand, convincing model predictions suggest that the hypotheses that were chosen are plausible so they can be used in the elaboration of an explanation (even if only an unrefined explanation) of the reality observed. On the other hand, the most immediate cause of a poor simulation of reality by a model can often be found in its underlying hypotheses which thus gain in being known. In addition, in more abstract lines, hypotheses are also important because they allow simplifications that may help formal developments. Furthermore, their very presence carries questions that can help orient the development and build a mathematical understanding of the theory that is considered. Thus, provided they be clearly expressed, hypotheses are precisely the model features that can be questioned and consequently that can allow a constructive comparison between reality and abstraction. In that, they represent one of the essential aspects of modelling, even (and perhaps especially) when they lack convincing modelling meaning.

Thus, hypotheses are, we believe, the principal primary foundations of modelling. They embody formally the sources of inaccuracy, incompleteness and possibly incorrectness on which the model is developed and on which relies the pertinence of its predictions. A set of ill-identified approximations represents a risk that the series of interpretations that are drawn from a model may slightly drift in a way that we have no knowledge of. It may thus be responsible for an orientation of our understanding of reality that eludes our control. In addition, more simply perhaps, by nature, hypotheses carry implicit assertions of facts that are not questioned unless the hypotheses themselves are. Because we have no guarantee that a question is not pertinent until it arises and is either answered or discarded deliberately, to avoid implicit answers, we believe that however subtle or reasonable they may be, hypotheses should be clearly explicated as well as rigorously analysed, so far as possible. And in particular, as argued in Section 4.2, special attention should be brought to the design of the theory and to the acknowledgement of the hypotheses that follow implicitly from its definition choices.

In summary, hypotheses not only are unavoidable embodiments of the impossibility of modelling a system (rather than a well-bounded set of its properties), they also are essential. They allow the approximations and simplifications of reality without which no modelling would be possible. And since they enclose their intrinsic incompleteness, they allow models to be effectively informative.

**Acknowledgments** We warmly thank K. Perrot for his pertinent comments. We also thank the *institut rhônalpin des systèmes complexes* (Ixxi) that supported the projet Maajes and the *réseau national des systèmes complexes* (RNSC) that supported the project Météding. The present article was partially fostered by both these projects.

## References

1. von Neumann, J.: Theory of self-reproducing automata. University of Illinois Press (1966)
2. Gardner, M.: Mathematical games: the fantastic combinations of John Conway's new solitaire game "life". Scientific American **223** (1970) 120–123
3. Goles, E.: Neural and automata networks: dynamical behaviour and applications. Kluwer Academic Publishers (1990)
4. Robert, F.: Les systèmes dynamiques discrets. Springer (1995)
5. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Journal of Mathematical Biology **5** (1943) 115–133
6. Kauffman, S.A.: Metabolic stability and epigenesis in randomly constructed genetic nets. Journal of Theoretical Biology **22** (1969) 437–467
7. Schelling, T.C.: Dynamic models of segregation. Journal of Mathematical Sociology **1** (1971) 143–186
8. Thomas, R.: Boolean formalisation of genetic control circuits. Journal of Theoretical Biology **42** (1973) 563–585
9. Ising, E.: Beitrag zur theorie des ferromagnetismus. Zeitschrift fur Physics **31** (1925) 253–258
10. Mandelbrot, B.B.: The fractal geometry of nature. W. H. Freeman (1983)
11. Thom, R.: Structural stability and morphogenesis. Addison-Wesley (1989)
12. Mendoza, L., Thieffry, D., Alvarez-Buylla, E.R.: Genetic control of flower morphogenesis in *Arabidopsis thaliana*: a logical analysis. Bioinformatics **15** (1999) 593–606
13. Mazoyer, J.: A six-state minimal time solution to the firing squad synchronization problem. Theoretical Computer Science **50** (1987) 183–228
14. Snoussi, E.H., Thomas, R.: Logical identification of all steady states: The concept of feedback loop characteristic states. Bulletin of Mathematical Biology **55** (1993) 973–991
15. Thomas, R., Thieffry, D., Kaufman, M.: Dynamical behaviour of biological regulatory networks—I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. Bulletin of Mathematical Biology **57** (1995) 247–276
16. Thieffry, D., Thomas, R.: Dynamical behaviour of biological regulatory networks—II. Immunity control in bacteriophage lambda. Bulletin of Mathematical Biology **57** (1995) 277–297
17. Richard, A.: Necessary conditions for multistationarity in discrete dynamical systems. Discrete Applied Mathematics **155** (2007) 2403–2413
18. Richard, A.: Positive circuits and maximal number of fixed points in discrete dynamical systems. Discrete Applied Mathematics **157** (2009) 3281–3288
19. Mendoza, L., Alvarez-Buylla, E.R.: Dynamics of the genetic regulatory network for *Arabidopsis thaliana* flower morphogenesis. Journal of Theoretical Biology **193** (1998) 307–319
20. Demongeot, J., Aracena, J., Thuderoz, F., Baum, T.P., Cohen, O.: Genetic regulation networks: circuits, regulons and attractors. Comptes Rendus Biologies **326** (2003) 171–188
21. Remy, É., Ruet, P., Thieffry, D.: Graphic requirement for multistability and attractive cycles in a Boolean dynamical framework. Advances in Applied Mathematics **41** (2008) 335–350
22. Demongeot, J., Goles, E., Morvan, M., Noual, M., Sené, S.: Attraction basins as gauges of robustness against boundary conditions in biological complex systems. PLoS One **5** (2010) e11793



23. Robert, F.: Discrete iterations: a metric study. Springer (1986)
24. Choffrut, C., ed.: Automata networks. Volume 316 of Lecture Notes in Computer Science. Springer (1988)
25. Weisbuch, G.: Complex systems dynamics: an introduction to automata networks. Perseus Books (1991)
26. Remy, E., Ruet, P.: From minimal signed circuits to the dynamics of Boolean regulatory networks. *Bioinformatics* **24** (2008) i220–i226
27. Siebert, H.: Dynamical and structural modularity of discrete regulatory networks. In: *COMP MOD*. Volume 6 of *EPTCS*. (2009) 109–124
28. Richard, A.: Local negative circuits and fixed points in non-expansive Boolean networks. *Discrete Applied Mathematics* **159** (2011) 1085–1093
29. Elena, A.: Algorithme pour la simulation dynamique des réseaux de régulation génétique. Master’s thesis, University J. Fourier (2004)
30. Tournier, L.: Étude et modélisation mathématique de réseaux de régulation génétique et métabolique. PhD thesis, Université Joseph Fourier – Grenoble 1 (2005)
31. Demongeot, J., Jézéquel, C., Sené, S.: Boundary conditions and phase transitions in neural networks. *Theoretical results. Neural Networks* **21** (2008) 971–979
32. Aracena, J., Goles, E., Moreira, A., Salinas, L.: On the robustness of update schedules in Boolean networks. *Biosystems* **97** (2009) 1–8
33. Goles, E., Noul, M.: Block-sequential update schedules and Boolean automata circuits. In: *Proceedings of Automata. DMTCS Proceedings* (2010) 41–50
34. Wilf, H.: *Generatingfunctionology*. Academic Press, NY
35. Sloane, N.A.J.: The on-line encyclopedia of integer sequences (OEIS), <http://oeis.org/A000670>
36. Tomic, P.: Modeling and analysis of the collective dynamics of large-scale multi-agent systems. PhD thesis, University of Illinois (2006)
37. Goles, E., Noul, M.: Disjunctive networks and update schedules. *Advances in Applied Mathematics* (2011) Accepted.
38. Noul, M.: General iteration graphs and Boolean automata circuits. Technical report, École normale supérieure de Lyon (2011)
39. Thomas, R.: Regulatory networks seen as asynchronous automata: A logical description. *Journal of Theoretical Biology* **153** (1991) 1–23
40. Harvey, I., Bossomaier, T.: Time out of joint: attractors in asynchronous random Boolean networks. In: *Proceedings of ECAL*, MIT Press (1997) 67–75
41. Bernot, G., Cassez, F., Comet, J.P., Delaplace, F., Müller, C., Roux, O.: Semantics of biological regulatory networks. *Electronic Notes in Computer Science* **180** (1997) 3–14
42. Bernot, G., Comet, J.P., Richard, A., Guespin, J.: Application of formal methods to biological regulatory networks: extending Thomas’ asynchronous logical approach with temporal logic. *Journal of Theoretical Biology* **229** (2004) 339–347
43. Richard, A.: Negative circuits and sustained oscillations in asynchronous automata networks. *Advances in Applied Mathematics* **44** (2010) 378–392
44. Delaplace, F., Kludel, H., Cartier-Michaud, A.: Discrete causal model view of biological networks. In: *Proceedings of CMSB*, ACM Press (2010) 4–13
45. Lam, S.S., Shankar, A.U.: A relational notation for state transition systems. *IEEE Transactions on Software Engineering* **16** (1990) 755–775
46. Clarke, E.M., Wing, J.M.: Formal methods: state of the art and future directions. *ACM Computing Surveys* **28** (1996) 626–643
47. Geurts, F.: Abstract compositional analysis of iterated relations: a structural approach to complex state transition systems. Springer (1998)

48. Cosnard, M., Demongeot, J.: On the definitions of attractors. In: *Iteration Theory and its Functional Equations*. Volume 1163 of *Lecture Notes in Mathematics*. Springer (1985) 23–31
49. Diner, S., Frague, D.: *Dynamical systems: A renewal of mechanism*. World Scientific (1987)
50. Kauffman, S.A.: *The origins of order*. Oxford University Press (1993)
51. Thomas, R.: On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations. In: *Numerical methods in the study of critical phenomena*. Volume 9 of *Springer Series in Synergetics*. Springer (1981) 180–193
52. Cook, S.A.: The complexity of theorem-proving procedures. In: *Proceedings of STOC*, ACM Press (1971) 151–158
53. Garey, M.R., Johnson, D.S.: *Computers and Intractability: guide to the theory of NP-Completeness*. Freeman (1979)
54. Noual, M.: Synchronism vs asynchronism in Boolean networks. Technical report, *École normale supérieure de Lyon* (2011)
55. Liang, S., Fuhrman, S., Somogyi, R.: Reveal, a general reverse engineering algorithm for inference of genetic network architecture. In: *Proceedings of PSB*. (1998) 18–29
56. Tournier, L., Chaves, M.: Uncovering operational interactions in genetic networks using asynchronous Boolean dynamics. *Journal of Theoretical Biology* **260** (2009) 196–209
57. Fatès, N., Thierry, É., Morvan, M., Schabanel, N.: Fully asynchronous behavior of double-quietest elementary cellular automata. *Theoretical Computer Science* **362** (2006) 1–16
58. Regnault, D., Schabanel, N., Thierry, É.: Progresses in the analysis of stochastic 2D cellular automata: a study of asynchronous 2D Minority. In: *Proceedings of MFCS*. Volume 4708 of *Lecture Notes in Computer Science*., Springer (2007) 320–332
59. Rouquier, J.B., Morvan, M.: Coalescing cellular automata: synchronization by common random source for asynchronous updating. *Journal of Cellular Automata* **4** (2009) 55–78
60. Thomas, R.: Logical vs. continuous description of systems comprising feedback loops: the relation between time delays and parameters. *Studies in Physical and Theoretical Chemistry* **8** (1983) 307–321
61. Thomas, R., d’Ari, R.: *Biological feedback*. CRC Press (1990)
62. Kaufman, M., Urbain, J., Thomas, R.: Towards a logical analysis of the immune response. *Journal of Theoretical Biology* **114**(4) (1985) 527–561
63. Siebert, H., Bockmayr, A.: Temporal constraints in the logical analysis of regulatory networks. *Theor. Comput. Sci.* **391** (2008) 258–275
64. Jacob, F., Monod, J.: Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology* **3** (1961) 318–356
65. Crick, F.: Central dogma of molecular biology. *Nature* **227** (1970) 561–563