

# Modal hypothesis logic, Boolean dynamical systems and genetic networks

Pierre Siegel, Andrei Doncescu, Vincent Risch, Sylvain Sené

Université publique, France

## Abstract

Genetic networks are biological systems that represent how genes or proteins interact in a cell. They are especially studied by means of automata networks and Boolean dynamical systems (BDSs). This article studies the representation of BDSs, using a *modal hypothesis logic*, namely  $\mathcal{H}$ . In the BDS formalism, a genetic network can be represented either by an *interaction graph* (IG) or by a *transition graph* (TG). Each of these representations stress distinct characteristics. The *dynamics* of a BDS is characterized by a function  $f$  and an updating mode  $\mu$  that organizes the entities updates over time. An important part of the studies done on BDSs focused on the analysis of both the stable configurations, or fixed points, of  $(f, \mu)$ , and stable/unstable cycles of  $(f, \mu)$ . The representation of a BDS by whatever default logic, ASP, or other nonmonotonic formalisms, enables to find stable configurations. However, these representations are not suitable to capture cyclic dynamical behaviors. We introduce representations for both IGs and *asynchronous* TGs in  $\mathcal{H}$ , which leads to new formal results. They aim at making possible to discriminate between stable configurations, limit cycles and unstable cycles. While a previous work has studied in detail IGs, the present paper focuses mainly on TGs, for which *ghost extensions*, defined in  $\mathcal{H}$ , play a key role.

*Keywords:* Modal logic, hypothesis logic, Boolean dynamical systems, genetic networks.

## 1 Introduction

From a logical point of view, a biological system can be viewed as a set of interacting elements, let's say entities, whose states change over discrete time. Genetic networks are specific biological systems that represent how the genes (or proteins) of a cell interact with each other for the survival, reproduction, or death of this cell. The study of genetic networks is a source of relevant questions regarding knowledge representation. First, interactions appear as a form of causality. As such, we expect to model it thanks to logical inferences, but of which kind? The use of classical logic is inadequate in this context because it cannot deal with inconsistencies, whereas what we learn on genetic networks arises often from long and expensive experiments and we know only a small part of the interactions while this knowledge can be revisable, uncertain, contradictory and even false. Moreover, algorithmic complexity is a crucial issue regarding the need to provide algorithms with reasonable calculation times in practice. These questions have been studied in artificial intelligence since the late 1970s, especially by the use of both particular nonmonotonic logics and techniques derived from constraint

programming. Notably, default logic (DL) [20, 6] as well as answer set programming (ASP) [16] can be used here.

Genetic networks have been studied from the end 1960s in the context of automata networks and Boolean dynamical systems (BDSs) as a set of entities mapped to Boolean states. In this framework, it is considered that the expression of one gene modulates the expression of another gene by activation or inhibition. The use of BDSs leads to founding theorems on feedback circuits, simply called circuits hereafter, that create behavioral complexity and richness [9, 17, 21, 22, 23, 24, 26, 33].

This article deals with the representation of BDSs, using a non-monotonic modal logic called *hypothesis logic* ( $\mathcal{H}$ ) [28, 30] defined in 1993, after a first approach proposed by [3]. In the context of BDSs, a genetic network can be represented by both an *interaction graph* (IG) and a *transition graph* (TG). The relationship between IGs and TGs have been studied since many years in dynamical systems theory, but remains an important open question. Our logic-based approach is a step toward a global clarification of this relationship. Preliminary results on circuits were given in [29].

A *dynamics* of a BDS is characterized by a function  $f$  associated with an updating mode  $\mu$  that organizes the entities updates over time (in this paper, we focus solely on the asynchronous one). Most of the studies done on BDSs have focused on their temporal asymptote, *i.e.*, on the analysis of both their stable configurations (or fixed points) and stable/unstable cycles. If the representation of a BDS by whatever DL, ASP or other well known nonmonotonic formalisms enables us to find fixed points, these representations are not suitable to capture cyclic dynamical behaviors. This is embarrassing because these cycles may represent real fundamental phenomena in living organisms such that the cell cycle [5, 15], the circadian cycle [1, 27], or the cardio-respiratory pace [10]. This possible lack of extensions in DL has been fully studied in the context of hypothesis logic. As shown in [30, 28], DL is a fragment of  $\mathcal{H}$ . In the latter logic, theories always have extensions among which some of them, called *ghost extensions*, have no counterpart in DL. And it is these ghost extensions that enable  $\mathcal{H}$  to discriminate between BDSs stable configurations, stable cycles and unstable cycles.

This article is structured as follows: Section 2 (resp. Section 3) gives the main definitions and notations related to  $\mathcal{H}$  (resp. BDSs); in Section 4, we present the way to represent BDSs in the hypothesis logic framework and develop the main results of the work showing that the asynchronous asymptotic behaviors such as stable configurations and stable cycles, as well as unstable cycles, are properly captured; Section 5 gives a brief conclusion on the work led.

## 2 Hypothesis Logic

**Syntax** Hypothesis logic  $\mathcal{H}$  [28, 30] is a bi-modal logic [4] with two modal operators  $L$  and  $[H]$ . If  $f$  is a formula, the intuitive meaning of  $Lf$  is *f is proved/stated*. The dual  $H$  of  $[H]$  is defined as  $Hf = \neg[H]\neg f$ . The intuitive meaning of  $Hf$  is *f is a hypothesis*, and hence  $[H]f$  means  *$\neg f$  is not a hypothesis*. The language of  $\mathcal{H}$ , denoted by  $\mathcal{L}(\mathcal{H})$ , is defined by the following inductive rules:

- Any formula of first-order logic is in  $\mathcal{L}(\mathcal{H})$ .
- Whenever  $f$  and  $g$  are in  $\mathcal{L}(\mathcal{H})$ ,  $\neg f$ ,  $(f \wedge g)$ ,  $(f \vee g)$ ,  $(f \rightarrow g)$ ,  $Lf$ ,  $[H]f$ ,  $Hf$  are in  $\mathcal{L}(\mathcal{H})$  too.

And no other formulas are in  $\mathcal{L}(\mathcal{H})$  than those formed by applying these two rules. Operator  $L$  has the properties of the modal system  $T$  and  $[H]$  has those of the modal system  $K$ . As a consequence, the inference rules and axiom schemata of  $\mathcal{H}$  are:

- All inference rules and axiom schemata of first-order logic.
- (N[H]):  $\vdash f \implies \vdash [H]f$ , the *necessitation rule* for [H].
- (NL):  $\vdash f \implies \vdash Lf$ , the *necessitation rule* for L.
- (K[H]):  $\vdash [H](f \rightarrow g) \rightarrow ([H]f \rightarrow [H]g)$ , the *distribution axiom schema* for [H].
- (KL):  $\vdash L(f \rightarrow g) \rightarrow (Lf \rightarrow Lg)$ , the *distribution axiom schema* for L.
- (TL):  $\vdash Lf \rightarrow f$ , the *reflexivity axiom schema* for L.

Unlike L, the axiom of reflexivity does not hold for [H]. There are so far no connections between L and [H]. We force this connection by adding the following *link axiom schema*:

- (LI):  $\vdash \neg(Lf \wedge H\neg f)$ .

This very weak axiom is one of the bases of  $\mathcal{H}$ . It means that it is impossible to prove  $f$  and to assume the hypothesis  $\neg f$  at the same time. Note the following equivalences:

$$\neg(Lf \wedge H\neg f) \iff Lf \rightarrow \neg H\neg f \iff H\neg f \rightarrow \neg Lf,$$

where the second (resp. third) formula means that *if we prove  $f$ , we cannot assume the hypothesis  $\neg f$*  (resp. *if we assume the hypothesis  $\neg f$ , we cannot prove  $f$* ).

**Semantics** As shown in [28],  $\mathcal{H}$  has a Kripke semantics with two accessibility relations,  $R[H]$  for [H],  $RL$  for L.  $R[H]$  is the relation of system  $K$  and  $RL$  is the relation of system  $T$ , hence reflexive. The relationship between the two relations, given by the link axiom, is  $RL \subseteq R[H]$ . Proofs of completeness, correctness, and compactness for  $\mathcal{H}$  are given in [28].

## 2.1 Hypothesis theories and extensions

As defined above,  $\mathcal{H}$  is a monotonic logic. In order to deal with the *revisable* character of usual informations (here of biological nature), a notion of *extension* is added just as in DL. However, contrary to the latter, three kinds of extensions are considered here, namely stable extensions, ghost extensions and sub-extensions.

**Definition 1.** *Given  $\mathcal{H}$ :*

- A hypothesis theory is a pair  $\mathcal{T} = \{HY, F\}$ , where  $F$  is a set of formulas of  $\mathcal{H}$  and  $HY$  is a set of hypotheses.
- An extension  $E$  of  $\mathcal{T}$  is a set  $E = \text{Th}(F \cup HY')$ , such that  $HY'$  is a maximal subset of  $HY$  consistent with  $F$ .
- A sub-extension  $E$  of  $\mathcal{T}$  is a set  $E = \text{Th}(F \cup HY')$ , such that  $HY'$  is a non-maximal subset of  $HY$  consistent with  $F$ .
- $E$  is a stable extension if it is an extension that satisfies the coherence property:  $\forall Hf, \neg Hf \in E \implies L\neg f \in E$ . Thanks to the link axiom schema, we hence get:  $\forall f, L\neg f \in E \iff \neg Hf \in E$ .
- $E$  is a ghost extension if it is an extension such that  $\exists Hf, \neg Hf \in E$  and  $L\neg f \notin E$ . Hence for a ghost extension, we only get:  $\forall f, L\neg f \in E \implies \neg Hf \in E$ .

**Definition 2.** *Let  $E$  be an extension or a sub-extension:*

1.  $E$  is complete if, for all  $i \in V$ ,  $Hi \in E$  or  $H\neg i \in E$ .
2. A propositional variable  $i \in V$  is free in  $E$  if  $Li \notin E$  and  $L\neg i \notin E$ . It is fixed otherwise.
3. The degree of freedom of  $E$ , denoted by  $\text{deg}(E)$ , is the number of free propositional variables that compose it.

It is shown in [30, 28] that, given a hypothesis theory  $\mathcal{T} = \{\text{HY}, \text{F}\}$ , if  $\text{F}$  is consistent then  $\mathcal{T}$  has at least one extension. Thus, an extension is obtained by adding one of the largest consistent sets of hypotheses to  $\text{F}$  while remaining consistent. Intuitively,  $E$  is stable if whenever it is forbidden to assume the hypothesis  $f$ ,  $\neg f$  is proven. It is a ghost extension otherwise. As shown in [28], stable extensions correspond to the standard extensions of DL (and to stable models of ASP). Ghost extensions do not have any correspondence in DL nor in ASP.

## 2.2 Example: representing genetic networks into $\mathcal{H}$

A *genetic network* represents interactions among genes or proteins in cell [2, 7, 8, 11, 13, 19]. From now on, let us consider that the entity at stake are proteins. In a modeling context, a protein is classically represented by an integer  $i \in \{1, \dots, n\}$ . Its *concentration* in a cell is denoted by  $x_i$ . In such networks, given a protein  $i$ , a set of interactions (or influences) from a set of proteins toward  $i$  describes in which conditions the concentration of  $i$  evolves. In the most general case, a concentration  $x_i$  is a real number. Here, we study the particular case where the concentrations  $x_i$  are in  $\{0, 1\}$ . This simplification may seem crude at first glance, but it actually makes sense because, experimentally, it is almost impossible to get precise concentrations. Genetic networks can be studied with the formalism of *Boolean dynamical systems* (BDSs), defined in the following section. To introduce our example, it suffices to know that, for a BDS, the concentration  $x_i = 1$  (resp.  $x_i = 0$ ) denotes the presence (resp. the absence) of protein  $i$  in the cell. Moreover, to study a BDS from a logical point of view, we just consider the congruence  $(x_i = 1) \equiv i$  (resp.  $(x_i = 0) \equiv \neg i$ ), viewing  $i$  as a Boolean variable.

One of the interests of hypothesis logic is that this bi-modal logic enables us to use three kinds of information:  $i$ ,  $\text{L}i$  and  $\text{H}i$ . Hence, by combining modalities with negations, we can use  $\{i, \text{H}i, \text{H}\neg i, \text{L}i, \text{L}\neg i\}$ . Remark that in  $\mathcal{H}$ , we have:  $\text{L}i \neq \neg\text{L}\neg i$ ,  $\neg\text{L}i \neq \text{L}\neg i$ ,  $\text{H}i \neq \neg\text{H}\neg i$  and  $\neg\text{H}i \neq \text{H}\neg i$ . This increasing of expressiveness allows for a more precise representation of biological networks. Let us consider the genetic network of a cell, and  $i$  a protein. Using hypothesis logic, we state that:

- $i$  means that  $i$  is present in the cell and  $\neg i$  means that it is absent.
- $\text{L}i$  means that  $i$  is produced by the cell ( $i$  is being activated) and  $\neg\text{L}i$  means that  $i$  is not produced ( $i$  is not being activated).
- $\text{L}\neg i$  means that  $i$  is destroyed by the cell ( $i$  is being inhibited) and  $\neg\text{L}\neg i$  means that  $i$  is not destroyed ( $i$  is not being inhibited).
- $\text{H}i$  (resp.  $\neg\text{H}i$ ) means that the cell gives (resp. does not give) the permission for attempting to produce  $i$ . In other words, the cell has (resp. has not) the ability to activate  $i$ .
- $\text{H}\neg i$  (resp.  $\neg\text{H}\neg i$ ) means that the cell gives (resp. does not give) the permission for attempting to destroy  $i$ . In other words, the cell has (resp. has not) the ability to inhibit  $i$ .

Regarding the use of  $\mathcal{H}$  in this context, the role of an extension appears to gather a maximum of consistent permissions. Note that even if  $\text{H}i$  stands for the cell giving permission to attempt the production of  $i$ , this production is not mandatory. It can be carried out or not, according to the context (*i.e.*, the set of all interactions in the cell). Similarly  $\text{H}\neg i$  gives the authorization to destroy  $i$ . It is important to note that  $\text{L}i$  and  $\text{L}\neg i$  are actually actions (production or destruction of a protein). So there is a difference between  $\text{L}\neg i$  which says that  $i$  is destroyed, and  $\neg\text{L}i$  which says that  $i$  is not produced, and hence is weaker. There is a similar distinction between  $\text{H}\neg i$  and  $\neg\text{H}i$  (and between

$Li$  and  $\neg L\neg i$ ; and between  $Hi$  and  $\neg H\neg i$ ).

Proposition below give some general properties of  $\mathcal{H}$ , particularly adequate for the modeling of the different states of proteins in a cell.

**Proposition 1.** *Given  $i$  a protein, the following results hold in  $\mathcal{H}$ :*

- (1)  $Li \rightarrow i$  and  $L\neg i \rightarrow \neg i$  (i.e., if  $i$  is produced (resp. destroyed), then  $i$  is present (resp. absent)).
- (2)  $\neg(Li \wedge H\neg i)$  and  $\neg(L\neg i \wedge Hi)$  (i.e., it is impossible to produce (resp. destroy)  $i$  and to give the permission to destroy (resp. produce)  $i$  it at the same time).
- (3)  $\neg(Li \wedge L\neg i)$  (i.e., it is impossible to produce and destroy  $i$  at the same time).

**Idea of the proof.** Axioms of  $\mathcal{H}$  are all what is needed. (1) are instances of axiom (T); (2) are instances of the linking axiom (LI); (3) derives directly from (1).

### 3 Boolean dynamical systems

A finite *Boolean dynamical system* (BDS) describes the evolution of the interactions in a network of  $n$  entities numbered from 1 to  $n$ , over discrete time. Consider  $V = \{1, \dots, n\}$  a set of  $n$  entities. A *configuration*  $x = (x_1, \dots, x_n)$  of the network is an assignment of a truth value  $x_i \in \{0, 1\}$  to each element  $i$  of  $V$ . The set of all configurations (i.e., all interpretations on the logic side), also called the *configuration space*, is denoted by  $X = \{0, 1\}^n$ . A *dynamics* of such a network is modeled via both a function  $f$ , called the *global transition function*, and an *updating mode*  $\mu$  that defines how the elements of  $V$  are updated over time. More formally,  $f : X \rightarrow X$  is such that  $x = (x_1, \dots, x_n) \mapsto f(x) = (f_1(x), \dots, f_n(x))$ , where each function  $f_i : X \rightarrow \{0, 1\}$  is a *local transition function* that gives the evolution of the entity  $i$  over time. There exists an infinite number of updating modes<sup>1</sup> among which the *parallel* and the *asynchronous* ones remain the most used [12, 32]. The *parallel*, or *perfectly synchronous*, updating mode is such that all the entities of the network are updated at each time step. Conversely, the *asynchronous* updating mode is a non-deterministic variation in which only one entity is updated at a time. In the sequel, we restrict our study to asynchronous dynamics [21, 24].

#### 3.1 Asynchronous transition graphs

Let  $X = \{0, 1\}^n$  be a space of configuration and  $f : X \rightarrow X$  the global transition function of a BDS. The *asynchronous dynamics* of  $f$  is given by its asynchronous transition graph (ATG)  $\mathcal{G}(f) = (X, T(f))$ , a digraph whose vertex set is the configuration space and arc set is the set of asynchronous transitions such that:  $T(f) = \{(x, y) \in X^2 \mid x \neq y, x = (x_1, \dots, x_i, \dots, x_n), y = (x_1, \dots, x_{i-1}, f_i(x), x_{i+1}, \dots, x_n)\}$ . Therefore, if  $(x, y) \in T(f)$ ,  $x$  and  $y$  differ exactly by one element; the transition is *unitary*.

**Note 1.** By definition, we relate a unique ATG  $\mathcal{G}(f)$  to a given function  $f$ . Moreover, by construction of ATGs,  $\mathcal{G}(f) = \mathcal{G}(g)$  whenever  $f$  and  $g$  are equivalent (i.e.,  $f(x) = g(x)$  for every  $x$ , or again  $f$  and  $g$  have the same truth tables). In other words, functions, truth tables and ATGs are isomorphic representations.

<sup>1</sup>Infinite, because deterministic updating modes are basically defined as infinite sequences of subsets of nodes of the network.

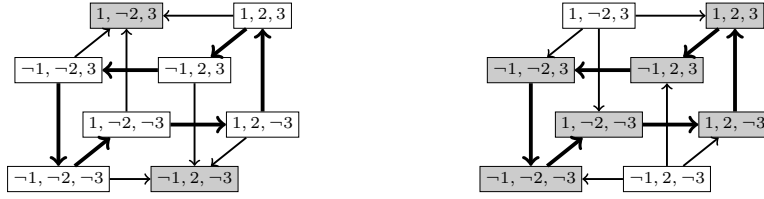


Figure 1: (left) ATG of function  $f$ , and (right) ATG of function  $g$ , presented in Example 1.

An *orbit* in  $\mathcal{G}(f)$  is a sequence of configurations  $(x^0, x^1, x^2, \dots)$  such that either  $(x^t, x^{t+1}) \in T(f)$  or  $x^{t+1} = x^t$  if  $x^t = f(x^t)$  (i.e.,  $x^t$  has no successors). A *cycle* of length  $r$  is a sequence of configurations  $(x^1, \dots, x^r, x^1)$  with  $r \geq 2$  whose configurations  $x^1, \dots, x^r$  are all different. From this, we derive what is classically called an asynchronous attractor in dynamical systems. An *attractor* is terminal *strongly connected component* (SCC) of  $\mathcal{G}(f)$ , i.e., a SCC with no outward transitions. Among attractors, we distinguish stable configurations from stable cycles. A *stable configuration* is a trivial attractor, i.e., a configuration  $x$  such that  $\forall i \in V, x_i = f_i(x)$ , which implies that  $x = f(x)$ . A *stable cycle* is a cyclic attractor such that, in  $\mathcal{G}(f)$ ,  $\forall t < r$ ,  $x^{t+1}$  is the unique successor of  $x^t$  and  $x^1$  is the unique successor of  $x^r$ . If an attractor is neither trivial nor cyclic, it is called a *stable oscillation*. When it is possible to get out from a SCC, this SCC is called an *unstable cycle* or an *unstable oscillation* depending on whether it is cyclic or not. An orbit that reaches a stable configuration stays there endlessly. Similarly, when it reaches a stable cycle, it adopts endlessly a stable oscillating behavior. Notice that in Figures, recurring configurations, i.e., configurations belonging to an attractor, are pictured in gray, and cycles are represented by bold transitions.

**Example 1. Boolean circuits of size 3** Consider  $V = \{1, 2, 3\}$ ,  $x = \{0, 1\}^3$  and two functions  $f$  and  $g$  such that  $f(x_1, x_2, x_3) = (\neg x_2, \neg x_3, x_1)$  and  $g(x_1, x_2, x_3) = (\neg x_3, x_1, x_2)$ . From the definitions of  $f$  and  $g$ , it is easy to derive their related ATGs,  $\mathcal{G}(f)$  and  $\mathcal{G}(g)$ , pictured in Figure 1. A transition corresponds to one arrow in the picture. There are up to 3 transitions leaving each configuration. Here,  $\mathcal{G}(f)$  has two symmetric stable configurations,  $(-1, 2, -3)$  and  $(1, -2, 3)$  while all the other configurations belong to an unstable cycle;  $\mathcal{G}(g)$  has a stable cycle, of length 6. This cycle is stable because there is only one transition leaving from each configuration, which is not the case for the unstable cycle of  $\mathcal{G}(f)$ . We will see in Section 4 that the two stable configurations of  $\mathcal{G}(f)$  correspond to two stable extensions of  $\mathcal{H}$ , and that the stable cycle of  $\mathcal{G}(g)$  corresponds to a set of 6 ghost extension of degree 1. The functions  $f$  and  $g$  can also be represented by *elementary circuits*, pictured in Figure 3. These ones are special cases of *interaction graphs*, defined below.

**Example 2.** Consider function  $h(x_1, x_2) = (\neg x_1 \vee x_2, x_1 \vee x_2)$  whose ATG is pictured on the left in Figure 2. This ATG has a stable state  $(1, 2)$  and an unstable cycle  $\{(-1, 2), (-1, -2)\}$ . There is an infinity of possible orbits because one can go follow the unstable cycle indefinitely, before attending  $(1, -2)$  and then stabilizing in  $(1, 2)$ .

**Example 3.** Consider function  $k(x_1, x_2) = (x_2, x_1 \wedge \neg x_1 \wedge x_2)$ , whose ATG is pictured on the right in Figure 2. This ATG has a stable state  $\{-1, -2\}$  and no cycles.



Figure 2: (left) ATG of function  $h$  presented in Example 2, and (right) ATG of function  $k$  presented in Example 3.

### 3.2 Interaction graphs and circuits

A transition graph is a very precise tool for studying the behavior of a function, but its size is exponential depending on the number of entities. Regarding practical applications, the information is often represented by more compact and more readable graphs of a different type. It is indeed the case with biological data which come from experiments that generally simply yield correlations among gene expressions. For BDSs, it is common to use interaction graphs.

An *interaction graph (IG)* is a signed digraph  $G = (V, I)$ , where  $V = \{1, \dots, n\}$  is the vertex set corresponding to the so called entities, and  $I \subseteq V \times S \times V$ , with  $S = \{-, +\}$  is the arc set corresponding to the so called interactions. An arc  $(i, +, j)$  (resp.  $(i, -, j)$ ) is said to be *positive* (resp. *negative*). From a dynamical point of view, the presence of an arc  $(i, s, j)$  in an IG means that the value of  $i$  affects the value of  $j$ , positively or negatively according to  $s$ : we say that  $i$  *regulates*  $j$ .

A *circuit*  $C = \{(i_1, s_{(1,2)}, i_2), \dots, (i_k, s_{(k,1)}, i_1)\}$  of size  $k$  is *elementary* if all the  $i_s$  that compose it are distinct. A circuit is *positive* (resp. *negative*) if it contains an even (resp. an odd) number of negative arcs.

Consider the toy example where  $j$  has only one incoming arc from  $i$ . In this case, the effect of the regulation is obvious: if the arc is positive (resp. negative),  $j$  will take the value (resp. the opposite value) of  $i$  after one update. Remark that elementary circuits are regulated this way.

**Note 2.** Consider an IG that contains an arc  $(i, s, i)$ , *i.e.*, a loop on  $i$ . If  $s = +$  (resp.  $s = -$ ), this arc makes  $i$  tend to maintain (resp. negate) its state. It depends of course on whether  $i$  admits other in-neighbors than itself or not, and on the positive or negative influence of these eventual neighbors. In the case that  $i$  admits no other in-neighbors, it is trivial that  $i$  endlessly maintains (resp. negate) its state if  $s = +$  (resp. if  $s = -$ ).

As mentioned above, an IG  $G = (V, I)$  represents the existence of the interactions involved between the entities of  $V$ . Specifying the nature of these interactions, and the conditions under which they occur effectively, leads to relate  $G$  to a BDS of function  $f$ . Then,  $G$  is the IG of  $f$  and is then denoted by  $G(f) = (V, I(f))$ . This is done by assigning a local transition function  $f_i$  to every  $i \in V$  so that  $\forall j \in V, \exists x \in \{0, 1\}^n, f_i(x) \neq f_i(\bar{x}^j) \iff (j, s, i) \in I(f)$ , where given  $x = (x_1, \dots, x_n)$ ,  $\bar{x}^j = (x_1, \dots, x_{j-1}, \neg x_j, x_{j+1}, \dots, x_n)$ . This specification induces the minimality of  $G(f)$  because each arc represents an effective interaction. It is also essential to note that if an IG corresponds to a single BDS/TG the converse is false. Contrary to TGs, an IG only provides static information about which entity acts which other and about the way it does.

**Example 4.** Figure 3 pictures the IGs associated with the ATGs of the BDSs defined from  $f$  and  $g$  in Example 1. Consider the positive circuit associated with  $f$  by following the directions of its arcs:



Figure 3: (*left*) the IG (a positive circuit) associated with ATG  $\mathcal{G}(f)$ , and (*right*) the IG (a negative circuit) associated with ATG  $\mathcal{G}(g)$  introduced in Example 1.

- Starting from  $x_1 = 1$ , or simply 1 using  $\mathcal{H}$ , we get the following infinite sequence:  $(1, 3, -2, 1, 3, -2, \dots)$ .
- Starting from  $x_1 = 0$ , or simply  $\perp 1$  using  $\mathcal{H}$ , we get the following infinite sequence:  $(-1, -3, 2, -1, -3, 2, \dots)$ .

The first (resp. second) dynamical behavior highlights the stable configurations  $1, 2, -3$  (resp.  $-1, -2, 3$ ) of  $f$ . For the negative circuit associated with  $g$ :

- Starting from  $x_1 = 1$ , or simply 1 using  $\mathcal{H}$ , we get the following infinite sequence:  $(-1, -2, -3, 1, 2, 3, -1, -2, -3, 1, 2, 3, \dots)$ .
- Starting from  $x_1 = 0$ , or simply  $\perp 1$  using  $\mathcal{H}$ , we get the following infinite sequence:  $(1, 2, 3, -1, -2, -3, 1, 2, 3, -1, -2, -3, \dots)$ .

In both cases, the observed dynamical behavior highlights the stable cycle of  $g$ .

### 3.3 General fundamental results and their biological direct applications

By considering that BDSs are good candidates for qualitatively modeling genetic networks (since established by the seminal papers [12, 32]), the presence of several attractors in their dynamical behaviors allows to model the cellular specialization. Indeed, if a genetic network controls a phenomenon of specialization, the cell will specialize (*i.e.*, will acquire a particular phenotype or a specific physiological function) according to the attractor towards which its underlying BDS evolves. A classical example of direct biological applications is the immunity control in bacteriophage  $\lambda$ , for which both lytic and lysogenic cycles of  $\lambda$  have been modeled in [31]. Another more tricky applications of BDSs in molecular systems biology concerns the floral morphogenesis of the plant *Arabidopsis thaliana* [18, 19]. Its dynamical behavior admits notably four stable configurations that correspond to the genetic expression patterns of the floral tissues, sepals, petal, stamens and carpels. This model has also allowed to formally explain the role of the hormone gibberellin on the floral development [8]. These works and the numerous other ones using BDSs or more general discrete dynamical systems (DDSs) emphasized the essential role of studies aiming at understanding the formal relations between IGs and TGs and their respective properties. They also clearly underlined the essential role of circuits, nowadays known as the behavioral complexity engines in dynamical systems. This comes in particular from Robert who established that, if the IG  $G(f)$  of a DDS  $f$  is acyclic, then  $f$  converges towards a unique stable configuration [25, 26]. Moreover, in [33], basing himself on asynchronous DDS, Thomas conjectured that  $G(f)$  of an asynchronous DDS  $f$  must contain a positive (resp. negative) circuit, for the latter to admit several stable configurations (resp. a non-trivial attractor such as a stable cycle or a stable oscillation). These two conjectures were proved to be true under the hypothesis of the asynchronous updating mode [21, 22, 23, 24].

Furthermore, in [21], the authors showed that an asynchronous positive (resp. neg-



ative) circuit of size  $n$  admits two attractors (resp. one attractor), namely two stable configurations  $x$  and its dual  $\bar{x}$  (resp. a stable cycle of length  $2n$ ). In [29], we obtained these results *via* the translation of the BDS into  $\mathcal{H}$ .

## 4 Representing BDS into $\mathcal{H}$

In [29], we studied in detail a translation of both positive and negative circuits into  $\mathcal{H}$ , which seems to be a first step to us because of their essential role in the regulation of the cell. In the sequel, we extend this translation to any asynchronous BDS. Indeed, our previous approach left formulas of the type  $(Hi \wedge Hj) \rightarrow Lk$  out of reach, whereas such formulas are essential for representing the notion of *binding* in genetic networks.

### 4.1 Syntax representation of BDS

Remind that an asynchronous BDS is characterized by a global transition function  $f : X \rightarrow X$  such that  $x = (x_1, \dots, x_n) \mapsto f(x) = (f_1(x), \dots, f_n(x))$ , where each function  $f_i : X \rightarrow \{0, 1\}$  is a local transition function. Also, remind that we consider that each  $x_i$  is a propositional variable  $i$ , that the assignment  $x_i$  is a Boolean value  $i$  or  $\neg i$ , and therefore that each  $f_i$  is a Boolean formula.

**Definition 3.** *The translation of a local transition function  $f_i$  into  $\mathcal{H}$  is given by a set  $TR(f_i)$  containing two formulas:  $TR(f_i) = \{Hf_i(x) \rightarrow Li \text{ and } H\neg f_i(x) \rightarrow L\neg i\}$ . The translation of a global transition function  $f : X \rightarrow X$  of a BDS in  $\mathcal{H}$  is the union of translations  $TR(f_i)$  for all  $i \in \{1, \dots, n\}$  such that  $TR(f) = \bigcup_{i=1}^n TR(f_i(x))$ .*

From the correspondence given in Note 1, this translation is equivalently the translation obtained for  $\mathcal{G}(f)$  and the truth tables of  $f$ .

**Example 5.** Consider  $V = \{1, 2, 3\}$ ,  $X = \{0, 1\}^3$ , and the global transition function  $f$  defined as  $f(x_1, x_2, x_3) = (\neg x_2, \neg x_3, x_1)$  of Example 1. Function  $f_1$  is translated into  $\mathcal{H}$  by  $TR(f_1) = \{H2 \rightarrow L\neg 1, H\neg 2 \rightarrow L1\}$ . Likewise  $f_2$  is translated by  $TR(f_2) = \{H3 \rightarrow L\neg 2, H\neg 3 \rightarrow L2\}$ , and  $f_3$  by  $TR(f_3) = \{H1 \rightarrow L3, H\neg 1 \rightarrow \neg L3\}$ . Therefore we obtain the following global translation:  $TR(f) = \{H2 \rightarrow L\neg 1, H\neg 2 \rightarrow L1, H3 \rightarrow L\neg 2, H\neg 3 \rightarrow L2, H1 \rightarrow L3, H\neg 1 \rightarrow \neg L3\}$  that admits two stable extensions<sup>2</sup>:

$E1 = Th(TR(f) \cup \{H1, H\neg 2, H3\})$  and  $E2 = Th(TR(f) \cup \{H\neg 1, H2, H\neg 3\})$ . When developing these extensions, we see that they are equivalent to their simplified forms:

- $E1 = \{H\neg 1, H2, H\neg 3, L1, L\neg 2, L3, \neg H1, \neg H\neg 2, \neg H3, \neg L\neg 1, \neg L2, \neg L\neg 3\}$ ;
- $E2 = \{H1, H\neg 2, H3, L\neg 1, L2, L\neg 3, \neg H\neg 1, \neg H2, \neg H\neg 3, \neg L1, \neg L\neg 2, \neg L3\}$ .

*In order to ease the reading and abusing notations, from now on in the text and in the figures, the extensions will contain only the  $Li$  and  $L\neg i$  that are true.* So, here,  $E1 = \{L1, L\neg 2, L3\}$  and  $E2 = \{L\neg 1, L2, L\neg 3\}$ . We can check that  $E1$  and  $E2$  are stable extensions (because for all  $i$ ,  $\neg Hi \in E1$  (resp.  $E2$ )  $\implies L\neg i \in E1$  (resp.  $E2$ ) and that  $E2$  is the mirror of  $E1$ ).

**Example 6.** Consider the global transition function  $g(x_1, x_2, x_3) = (\neg x_3, x_1, x_2)$  of Example 1. The translation in  $\mathcal{H}$  leads to the following set of formulas:  $F(G(g)) =$

<sup>2</sup>This is shown by attempting to add to  $F(G(f))$  each subset of  $HY(G(f))$  and keeping only those among them that are the maximals ones consistent with  $F(G(f))$ . This can be done very quickly using a SAT solver.

$\{H1 \rightarrow L2, H2 \rightarrow L3, H3 \rightarrow L^{-1}, H^{-1} \rightarrow L^{-2}, H^{-2} \rightarrow L^{-3}, H^{-3} \rightarrow L1\}$ . This allows us to obtain the following 6 equivalent ghost extensions:  $E1 = \{L2, L3\}$ ,  $E2 = \{L^{-1}, L3\}$ ,  $E3 = \{L^{-1}, L^{-2}\}$ ,  $E4 = \{L^{-2}, L^{-3}\}$ ,  $E5 = \{L1, L^{-3}\}$ ,  $E6 = \{L1, L2\}$ . The following statements hold:

- $E1, \dots, E6$  are extensions because they are maximal consistent. They are ghost extensions because in each of them there is a  $\neg Hi$  (resp.  $\neg H^{-i}$ ) without  $L^{-i}$  (resp.  $\neg Hi$ ).
- These extensions are of degree 1.
- In [29], we proved that there exists a permutation on the  $i$ s that allows us to pass from  $E1$  to  $E2, \dots, E5$  to  $E6$ . This permutation represents the stable cycle of  $g$ .

Moreover, there are also two sub-extensions,  $E7 = \{1, -2, 3\}$  and  $E8 = \{-1, 2, -3\}$  that contain neither  $Li$  nor  $L^{-i}$ . Hence all the  $i$ s are free and their degree is 3.

**Example 7.** Consider function  $k$  such that  $k(x_1, x_2) = (x_2, x_1 \wedge \neg x_1 \wedge x_2)$ , whose ATG is pictured in Figure 2. Function  $k_1$  is translated into  $\mathcal{H}$  by the couple  $TR(k_1) = \{H2 \rightarrow L^{-1}, H^{-2} \rightarrow L1\}$ . Function  $k_2$  is translated by  $TR(k_2) = \{H(1 \wedge \neg 1 \wedge 2 \rightarrow L2), H^{-1}(1 \wedge \neg 1 \wedge 2) \rightarrow L^{-2}\}$ . Since  $\neg(1 \wedge \neg 1 \wedge 2) = \neg 1 \vee 1 \vee \neg 2$ , we finally obtain the following global translation into  $\mathcal{H}$  for  $k$ :  $TR(k) = \{H2 \rightarrow L^{-1}, H^{-2} \rightarrow L1, H(1 \wedge \neg 1 \wedge 2) \rightarrow L2, H^{-1}(1 \wedge \neg 1 \wedge 2) \rightarrow L^{-2}\}$ , which admits three extensions:

- A stable extension  $E1 = Th(TR(k) \cup \{H^{-1}, H^{-2}\}) = \{L^{-1}, L^{-2}\}$ ;
- Two ghost extensions of degree 1:  $E2 = Th(TR(k) \cup \{H1, H^{-1}\}) = \{L^{-2}\}$ , and  $E3 = Th(TR(k) \cup \{H2\}) = \{L1\}$ .

**Note 3.** Function  $k$  may appear naive, because  $x_1 \wedge \neg x_1 \wedge x_2 = \perp$ , which gives an equivalent translation  $TR(h) = \{H2 \rightarrow L^{-1}, H^{-2} \rightarrow L1, H\perp \rightarrow L2, H\top \rightarrow L^{-2}\}$ . However, one of the aims of this study is also to show that we can deal with functions of any kind, without the need of a pre-processing. The formalism of  $\mathcal{H}$  and the algorithms, that can in particular use a SAT solver, implicitly make the expected simplifications.

## 4.2 Semantic representation of ATGs into $\mathcal{H}$

This section studies the formal relationship between ATGs and their representation into  $\mathcal{H}$ . It uses *Kripke semantics* [14] such as defined for *normal modal logics* (i.e., the logics that contain at least axiom (K)). We shortly remind here the bases needed for our developments. A *Kripke structure* is a digraph  $K = (W, R)$  where the *universe*  $W$  is a set  $\{w_1, \dots, w_n\}$  of *worlds* and the *accessibility relation*  $R \subseteq W \times W$  is a binary relation among worlds. When  $w_j R w_k$ ,  $w_k$  is *accessible* from  $w_j$ . A *Kripke model* is obtained by assigning in every world a truth value to every proposition  $i$ . This makes possible to assign a truth value to all the formulas of the propositional calculus (PC). A world is then mapped to a logical interpretation and hence implicitly to a configuration of a BDS. Formulas other than those of PC are assigned to worlds with the following condition: for all  $f$ ,  $Lf$  is true in a world  $w_k$  if and only if  $f$  is true in all accessible worlds from  $w_k$ . The different axioms that hold in different modal logics depend on the properties of the accessibility relations  $R$ . As well known, for the system  $K$ ,  $R$  is any relation, while axiom (TL) holds if and only if  $R$  is reflexive. In the following, we give a morphism between ATGs and Kripke models for the modal system  $T$ , which allows us to exhibit a morphism from hypothesis theories to ATGs.

Let  $V = \{1, \dots, n\}$  be a set of entities,  $X = \{0, 1\}^n$  be a configuration space,  $f : X \rightarrow X$  be a function with its associated ATG  $\mathcal{G}(f) = (X, T(f))$ . Remind that  $T(f)$  is a set of edges corresponding to transitions. We now look at an increased version of  $\mathcal{G}(f)$ , namely  $\mathcal{G}^*(f) = (X, T \cup \hookrightarrow)$  where  $\hookrightarrow$  denotes the reflexivity such that  $(x, x)$

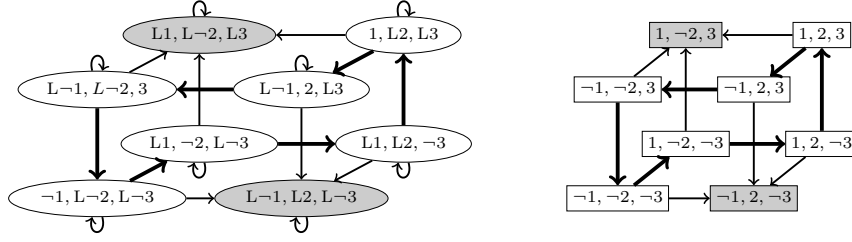


Figure 4: (left) Kripke model of function  $f$  given in Example 1, and (right) its associated ATG.

is a transition of  $\mathcal{G}^*(f)$  for all  $x \in \{0, 1\}^n$ , as a Kripke structure whose universe is  $X$  and whose accessibility relation is  $T(f) \cup \hookrightarrow$ . If we consider that any configuration  $x \in X$  is a world, we get a Kripke model with  $T(f) \cup \hookrightarrow$  as accessibility relation. Therefore, we get an isomorphism between Kripke models and increased ATGs, from which it is trivial to obtain the related ATGs. In order to obtain a morphism between hypothesis theories and ATGs, we define the concept of a *projection* of an extension or of a sub-extension.

**Definition 4.** Consider a sub-extension, or an extension,  $E$  of  $\mathcal{H}$ . The projection of  $E$  on the system  $T$  is the set of formulas of  $E$  that do not contain the operator  $H$ .

Now, if  $\mathcal{T} = \{HY, F\}$  is an hypothesis theory, and  $P$  is the set of the projections of the extensions or of the sub-extensions of  $\mathcal{T}$ , we obtain a morphism from  $\mathcal{T}$  to  $P$ , and therefore a morphism from hypothesis theories to Kripke models.

**Note 4.** Note that one does not get an isomorphism. Indeed the projections of two different extensions can be equal, and therefore be related to the same Kripke model. Note also that, since every edge connects two configurations that differ by one entity in an ATG, two different accessible worlds of the Kripke model differ by one propositional variable.

**Example 8.** On the left of Figure 4 is depicted the Kripke model associated with the projection of the ATG of function  $f$  given in Example 1. The eight nodes are the worlds, and the arcs express the accessibility relation. The two nodes in gray of the Kripke model,  $\{L1, L-2, L3\}$  and  $\{L-1, L2, L-3\}$  represent the two stable extensions from the translation of  $f$ . Their degree of freedom is 0. The other six nodes are sub-extensions since they are non-maximal. Their degree is 2. These six nodes form the unstable cycle of  $f$ , represented by bold transitions. We remark that the set of edges of the Kripke model contains the set of edges of the ATG. The missing edges are the loops related to the axiom of reflexivity. We also remark that in order to get the Kripke model from the ATG, it is enough to inject the modality  $L$  at the right places into the cube, according to the corresponding Kripke frame.

**Example 9.** On the left (resp. on the right) of Figure 5 is depicted the Kripke model (resp. the ATG) of function  $g$  given in Example 1. The six nodes

$$\{1, L2, L3\}, \{L-1, 2, L3\}, \{L-1, L-2, 3\}, \\ \{-1, L-2, L-3\}, \{L1, -2, L-3\}, \{L1, L2, -3\}$$

represent the ghost extensions of the translation of  $g$ . Their degree of freedom is 1.

The other two nodes  $\{1, -2, 3\}$  and  $\{-1, 2, -3\}$  are sub-extensions because they are non-maximal; their degree of freedom is 3. The six extensions of the Kripke model correspond to the six configurations of the stable cycle of function  $g$ .

**Example 10.** As in the two former examples, Figure 6 depicts both the Kripke model and the ATG of function  $k$  given in Example 3. Three nodes represent the three extensions of the translation of  $k$ : node  $\{L-1, L-2\}$  represents the stable one,  $\{1, L-2\}$  and  $\{L1, 2\}$  the two ghost ones of degree 1. There is also one sub-extension of degree 2,  $\{1, 2\}$ .

### 4.3 Results

**Proposition 2.** *Let  $\mathcal{T}$  be an hypothesis theory,  $E$  be an extension or a sub-extension of  $\mathcal{T}$ ,  $w$  be its projection and  $k$  be the degree of freedom of  $E$ . In the Kripke model associated to  $\mathcal{T}$ , there are exactly  $k$  distinct worlds, different from  $w$ , reachable from  $w$ .*

*Proof.* Since the degree of  $E$  is  $k$ , there are  $\{i_1, \dots, i_k\}$  propositional variables, free in  $E$ . For every  $i \in \{i_1, \dots, i_k\}$ , we have both  $\neg Li \in E$  and  $\neg L\neg i \in E$ . Two cases are possible, either  $i \in E$  or  $\neg i \in E$ . If  $i \in E$ , since  $\neg Li \in E$ , there exists a world  $w'$  accessible from  $w$ , and distinct from  $w$ , that contains  $\neg i$ . Regarding the second case, if  $\neg i \in E$ , since  $\neg L\neg i \in E$ , there exists a world  $w''$  accessible from  $w$ , and distinct from  $w$ , that contains  $i$ . Therefore, for each  $i \in \{i_1, \dots, i_k\}$ , there is a world accessible from  $w$ , and distinct from  $w$ , that contains the opposite of  $i$ . Because  $w$  is related to an asynchronous ATG, from Note 4, all these accessible worlds are distinct. Hence there are  $k$  distinct worlds reachable from  $w$ .  $\square$

**Theorem 1.** *Let  $\mathcal{G}(f)$  be an ATG of function  $f$ , and  $TR(f)$  be its associated hypothesis theory. The following holds:*

1. *If  $x = \{x_1, \dots, x_n\}$  is a stable configuration of  $\mathcal{G}(f)$ , then there exists an extension  $E$  of degree 0, issued from  $TR(f)$ , that contains  $\{Lx_1, \dots, Lx_n\}$ .*
2. *Let  $E$  be an extension of degree 0, issued from  $TR(f)$ , and  $w$  the projection of  $E$ . If  $x$  is the configuration related to  $w$ , then  $x$  is stable.*

*Proof.* Each statement is proved separately:

1. If  $x$  is a stable configuration of  $\mathcal{G}(f)$ , no edges can leave from  $x$ . By construction of the Kripke model, the same holds for the Kripke world  $w$  related to  $x$ . Hence the only world accessible from  $w$  is  $w$ , that is, for any  $i \in w$  (resp.  $\neg i \in w$ ),  $Li \in$

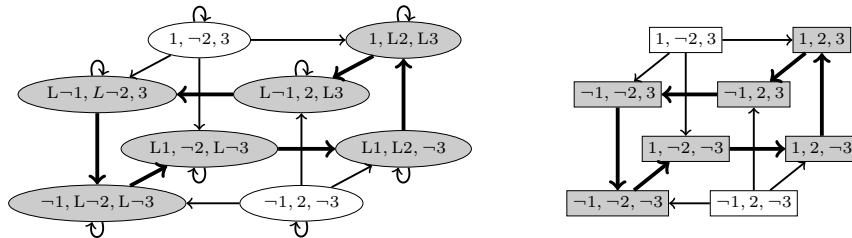


Figure 5: (left) Kripke model of function  $g$  given in Example 1, and (right) its associated ATG.

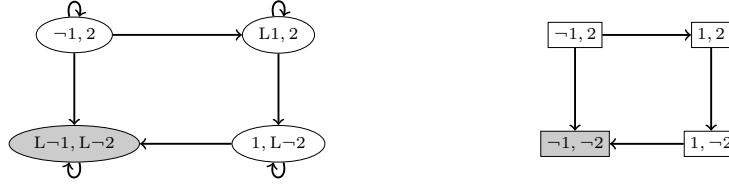


Figure 6: (left) Kripke model of function  $k$  given in Example 7, and (right) its associated ATG.

$w$  (resp.  $L-i \in w$ ). Therefore, every  $i$  is fixed and the degree of the extension  $E$ , issued from  $TR(f)$ , is 0.

2. Let the projection of  $E$  be represented by the world  $w$ . Since  $E$  is of degree 0, from Proposition 2, the only reachable world from  $w$  is  $w$ . By construction of the Kripke model, the same holds for  $x$ . Therefore  $x$  is a stable configuration of  $\mathcal{G}(f)$ . □

**Theorem 2.** Let  $\mathcal{G}(f)$  be the ATG of function  $f$  and  $TR(f)$  be its associated hypothesis theory. Every stable cycle  $C$  of  $\mathcal{G}(f)$  corresponds to a cycle of extensions of degree 1 in  $TR(f)$ .

*Proof.* The proof is similar to that of Theorem 1. Let  $C = \{x_1, \dots, x_k\}$  be a stable cycle of  $\mathcal{G}(f)$ , and  $W = \{w_1, \dots, w_\ell\}$  the set of extensions associated with  $C$ . By construction of the Kripke model,  $W$  is also a cycle of same length as  $C$ . Since  $C$  is stable, each of its configurations  $x_i$  admits only one outward arc. And the same property holds for  $w_i$ , i.e., the degree of  $w_i$  is 1. Therefore, all extensions of  $W$  are of degree 1. □

Analog theorems were proved in the context of [29]. They correspond to the results given in [21]. With the same arguments as those used for the proofs of the previous theorems, we can show that if a BDS contains an unstable cycle  $C$ , it is represented by a set of extensions such that at least one of those is of degree greater than 1. Indeed, if the cycle is unstable, it contains a configuration  $x$  of degree greater than 1 and, by construction, the Kripke model associated with the BDS contains the extension  $E$  corresponding to  $x$ .

## 5 Conclusion

In [29], we studied in detail a translation of both positive and negative asynchronous circuits into  $(\mathcal{H})$ . In this paper, we extend this translation to any asynchronous BDS, by showing that hypothesis logic capture some of their essential behavioural, such as stable configurations and stable cycles that are specific attractors and unstable cycles. Of course, these results pave the way to further studies about how hypothesis logic could enable to represent all the dynamical richness of BDSs, by taking for instance into account their stable and unstable oscillations and other known properties related to the orbits.

The lack of space prevents a serious discussion about algorithms in the context of this work. However our approach provides us with hints for simple algorithms that can

distinguish stable/unstable states, and as such, provides ways for enumerating all the solutions in a practical way. Eventually, note that an extension is obtained by adding a consistent maximal set of hypotheses. Since it is possible to test whether consistency is preserved when adding each hypothesis, the computation of extensions is non-deterministic but constructive, contrary to DL and ASP.

**Acknowledgement** This work was mainly funded by our salaries as French State agents or pensioner, affiliated to Université des Antilles, LAMIA, La Guadeloupe, France (AD) ; Université d’Aix-Marseille, Université de Toulon, CNRS, LIS, UMR 7020, Marseille, France (PS, VR, and SS); and secondarily by the project ANR-18-CE40-0002 FANs (SS).

## References

- [1] O. E. Akman, S. Watterson, A. Parton, N. Binns, A. J. Millar, and P. Ghazal. Digital clocks: simple Boolean models can quantitatively describe circadian systems. *Journal of The Royal Society Interface*, 9:2365–2382, 2012.
- [2] J. Aracena, M. González, A. Zuñiga, M. A. Mendez, and V. Cambiazo. Regulatory network for cell shape changes during *Drosophila* ventral furrow formation. *Journal of Theoretical Biology*, 239:49–62, 2006.
- [3] P. Besnard and P. Siegel. Supposition-Based Logic for Automated Nonmontonic Reasoning. In *Proceedings of CADE’88*, volume 310 of *LNCS*, pages 592–601. Springer, 1988.
- [4] B. Chellas. *Modal logic: an introduction*. Cambridge University Press, 1980.
- [5] M. I. Davidich and S. Bornholdt. Boolean network model predicts cell cycle sequence of fission yeast. *PLoS One*, 3:e1672, 2008.
- [6] J. P. Delgrande and T. Schaub. Expressing default logic variants in default logic. *Journal of Logic and Computation*, 15:593–621, 2005.
- [7] J. Demongeot, A. Elena, M. Noual, S. Sené, and F. Thuderoz. “Immunetworks”, intersecting circuits and dynamics. *Journal of Theoretical Biology*, 280:19–33, 2011.
- [8] J. Demongeot, E. Goles, M. Morvan, M. Noual, and S. Sené. Attraction basins as gauges of the robustness against boundary conditions in biological complex systems. *PLoS One*, 5:e11793, 2010.
- [9] J. Demongeot, M. Noual, and S. Sené. Combinatorics of Boolean automata circuits dynamics. *Discrete Applied Mathematics*, 160:398–415, 2012.
- [10] O. Dergacheva, K. J. Griffioen, R. A. Neff, and D. Mendelowitz. Respiratory modulation of premotor cardiac vagal neurons in the brainstem. *Respiratory Physiology & Neurobiology*, 174:102–110, 2010.
- [11] A. Fauré, A. Naldi, C. Chaouyia, and D. Thieffry. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22:e124–e131, 2006.

- [12] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed nets. *Journal of Theoretical Biology*, 22:437–467, 1969.
- [13] S. A. Kauffman, C. Peterson, B. Samuelsson, and C. Troein. Random Boolean network models and the yeast transcriptional network. *PNAS*, 100:14796–14799, 2003.
- [14] S. A. Kripke. Semantical analysis of modal logic I Normal modal propositional calculi. *Mathematical Logic Quarterly*, 9:67–96, 1963.
- [15] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang. The yeast cell-cycle network is robustly designed. *PNAS*, 101:4781–4786, 2004.
- [16] V. Lifschitz. Action languages, answer sets, and planning. In *The Logic Programming Paradigm: A 25-Year Perspective*, pages 357–373. Springer, 1999.
- [17] T. Melliti, M. Noual, D. Regnault, S. Sené, and J. Sobieraj. Asynchronous dynamics of Boolean automata double-cycles. In *Proceedings of UCNC'15*, volume 9252 of *LNCS*, pages 250–262. Springer, 2015.
- [18] L. Mendoza and E. R. Alvarez-Buylla. Dynamics of the genetic regulatory network for *Arabidopsis thaliana* flower morphogenesis. *Journal of Theoretical Biology*, 193:307–319, 1998.
- [19] L. Mendoza, D. Thieffry, and E. R. Alvarez-Buylla. Genetic control of flower morphogenesis in *Arabidopsis thaliana*. *Bioinformatics*, 15:593–606, 1999.
- [20] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.
- [21] É. Remy, B. Mossé, C. Chaouyia, and D. Thieffry. A description of dynamical graphs associated to elementary regulatory circuits. *Bioinformatics*, 19:ii172–ii178, 2003.
- [22] É. Remy, P. Ruet, and D. Thieffry. Graphic requirement for multistability and attractive cycles in a Boolean dynamical framework. *Advances in Applied Mathematics*, 41:335–350, 2008.
- [23] A. Richard. Negative circuits and sustained oscillations in asynchronous automata networks. *Advances in Applied Mathematics*, 44:378–392, 2010.
- [24] A. Richard and J.-P. Comet. Necessary conditions for multistationarity in discrete dynamical systems. *Discrete Applied Mathematics*, 155:2403–2413, 2007.
- [25] F. Robert. Itérations sur des ensembles finis et automates cellulaires contractants. *Linear Algebra and its Applications*, 29:393–412, 1980.
- [26] F. Robert. *Discrete Iterations: A Metric Study*. Springer, 1986.
- [27] T. Roenneberg and M. Meroow. The network of time: understanding the molecular circadian system. *Current Biology*, 13:R198–R207, 2003.
- [28] C. Schwind and P. Siegel. A modal logic for hypothesis theory. *Fundamenta Informaticae*, 21:89–101, 1994.

- [29] P. Siegel, A. Doncescu, V. Risch, and S. Sené. Towards a Boolean dynamical system representation into a nonmonotonic modal logic. In *Proceedings of NMR'18*, pages 53–62, 2018.
- [30] P. Siegel and C. Schwind. Modal logic based theory for non-monotonic reasoning. *Journal of Applied Non-classical Logic*, 3:73–92, 1993.
- [31] D. Thiéffry and R. Thomas. Dynamical behaviour of biological regulatory networks – II. Immunity control in bacteriophage Lambda. *Bulletin of Mathematical Biology*, 57:277–297, 1995.
- [32] R. Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42:563–585, 1973.
- [33] R. Thomas. 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*, pages 180–193. Springer, 1981.