Representation of Boolean Genetic Regulatory Networks by Hypothesis Logic

Pierre Siegel[1], Andrei Doncescu[2], Vincent Risch[1], Sylvain Sené[1,3]

[1] Aix Marseille Univ, CNRS, LIS, Marseille, France
[2] French West Indies University, Lamia, Guadeloupe
[3] Université publique, Marseille, France

Abstract. Boolean networks (BNs) are systems of entities described by Boolean variables providing interaction in discrete time. They are particularly used in the modeling of Cell Signaling called Boolean Genetic Regulation Networks (BGRN). We present in this article a new representation of BGNR. For this representation, we use a *modal Hypothesis Logic* ($\mathcal{H}$). We also introduce a representation for BNs in $\mathcal{H}$ which drives to new formal results. Our results make it possible to discriminate between stable configurations, limit cycles and unstable cycles of BNs. *Ghost Extensions*, defined in $\mathcal{H}$, play here a key role.

Keywords: Gene regulation, Genetic networks, Boolean networks, Modal logic, Hypothesis logic, Computational Systems Biology

1 Introduction

From a logical point of view, a biological system can be viewed as a set of interacting elements (or entities), whose states change over discrete time. Genetic networks are specific biological systems that represent how the genes/proteins of a cell interact with each other for the survival, reproduction, or death of this cell. They have been studied from the end of the 1960s in the context of Boolean networks (BNs), as sets of entities mapped to Boolean states, and their associated Boolean dynamical systems (BDSs). 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 that create behavioral complexity and richness [9, 19, 23–26, 28, 35].

The study of genetic networks can be a source of relevant questions regarding knowledge representation. Firstly, interactions appear as a form of causality. Therefore, we expect that the models use logical inferences, but which inference is the most adequate for the genetic networks?

Of course, the use of classical logic is inadequate in this context because it cannot deal with inconsistencies, whereas what we learn on genetic networks often arises from long and expensive experiments, and even then we only know or observe a small part of the interactions. This knowledge could 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) [22, 6] as well as answer set programming (ASP) [18] are good approaches. For the representation of genetic networks with no cycle, results have been obtained with default logic [12, 11]. This article deals with a representation of boolean genetic networks and of a representation of BDSs using a non-monotonic modal logic called *hypothesis logic* ($\mathcal{H}$) [30, 32] defined in 1993, after a first approach proposed by [3]. Preliminary results on circuits were presented in [31].

A *dynamic* of a BDS is characterized by a function f associated with an updating mode μ that organizes the entities updates over time. In this paper, we focus solely on the fully asynchronous (non-deterministic) updating mode. Most of the studies done on BDSs have focused on their temporal asymptotic, *i.e.*, on the analysis of both their stable configurations (or fixed points) and stable/unstable cycles/oscillations. 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, 17], the circadian cycle [1, 29], 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 [32, 30], 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. We introduce representations in $\mathcal{H}$ for BN asynchronous transition graphs (ATGs), which allows us to exhibit new formal results. Notably, ghost extensions play here a key role since they enable $\mathcal{H}$ to discriminate between BDSs stable configurations, stable and unstable attractors. This article is structured as follows: Section 2 reminds basic results about non-monotonicity logic. Section 3 gives the main definitions and notations related to $\mathcal{H}$. Section 4 gives a representation of genetic networks into $\mathcal{H}$. Section 5 gives the definitions on BDS. In Section 6, we present a syntax and a semantic representations of BDSs in $\mathcal{H}$. We also shows that the asynchronous asymptotic behaviors such as stable configurations and stable attractors, as well as unstable attractors, are properly captured. Section 7 gives a brief conclusion.

2 Non-monotonic and default logics

Representing biological system with a logical formalism seems intuitive. Such a representation from classical logic is not adapted because it leads to inconsistencies in most cases. A way to manage these inconsistencies is provided by *non-monotonic formalisms*. Monotonicity is a property of inference relations that states that deductions directly increases/decreases with knowledges: whereas this property is crucial in mathematics, it is largely questionable regarding reasoning with incomplete or contradictory information. That led to the development of *non-monotonic logics* in artificial intelligence among which *Default Logic* or ASPs. Default logic [22] concerns standard formulas of first order logic, to which contextual inference rules called defaults are added in order to deal with revisable informations. A *default* is a local inference rule $d = \frac{A\,:\,B}{C}$, whose application specifically depends on the formulas A, B, C that compound it. The intuitive meaning is: "If A holds, if B is coherent with what is known, then C holds". The fact that a default can be triggered or not depending on the context, further leads to a notion of *extensions* as max-consistent sets of formulas with respect to the trigger of the defaults used to get it. The underlying reasoning is non-monotonic because adding here a new information may invalidate previously triggered defaults. A first remark is that some default theories may have no extensions, which expresses a form of deep inconsistency which renders computation more tricky. A second remark is that default logic only computes stable extensions, which appears as a drawback. Indeed, such a type of extension is limited since it appears too cheap to handle more than stable attractors of BDSs while we actually expect to capture also unstable ones. Intuitively, the main problem arises from how the dynamics is represented.

3 Hypothesis Logic

Hypothesis logic $\mathcal{H}$ [30, 32] is a bimodal 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 $\mathrm{H}f = \neg[\mathrm{H}]\neg f$. The intuitive meaning of Hf is f *is a hypothesis*, and hence $[\mathrm{H}]f$ means $\neg f$ *is not a hypothesis*. For example, a default $\frac{A\,:\,B}{C}$ can be interpreted/translated in $\mathcal{H}$ by the modal formula $\mathrm{L}A \wedge \mathrm{H}B \rightarrow \mathrm{L}C$ whose intuitive meaning is: If A is stated and B is a valid hypothesis then C is stated. The formalism used in this article uses a restricted definition of the language of $\mathcal{H}$, sufficient to represent BNs and Boolean Genetic Networks.

3.1 Syntax

The language of $\mathcal{H}$, denoted by $\mathscr{L}(\mathcal{H})$, is defined by the following inductive rules:
- Any formula of propositional calculus (PC) is in $\mathscr{L}(\mathcal{H})$.
- The set of atoms (or propositional variables) of $\mathscr{L}(\mathcal{H})$ is finite.
- Whenever f and g are formulas of PC, Lf, $[\mathrm{H}]f$, Hf, $\neg$Lf, $\neg[\mathrm{H}]f$, $\neg$Hf are in $\mathscr{L}(\mathcal{H})$ too[4].

And no other formulas are in $\mathscr{L}(\mathcal{H})$ 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]): $\vdash f \Rightarrow \vdash [\mathrm{H}]f$, the *necessitation rule* for [H].

[4] The full definition of $\mathcal{H}$ further states that any formula of first-order logic is in $\mathscr{L}(\mathcal{H})$, and that, whenever f and g are in $\mathscr{L}(\mathcal{H})$, $\neg f$, $(f \wedge g)$, $(f \vee g)$, $(f \rightarrow g)$, are in $\mathscr{L}(\mathcal{H})$ too.

- (NL): $\vdash f \Rightarrow \vdash \mathrm{L}f$, the *necessitation rule* for L.
- (K[H]): $\vdash [\mathrm{H}](f \rightarrow g) \rightarrow ([\mathrm{H}]f \rightarrow [\mathrm{H}]g)$, the *distribution axiom schema* for [H].
- (KL): $\vdash \mathrm{L}(f \rightarrow g) \rightarrow (\mathrm{L}f \rightarrow \mathrm{L}g)$, the *distribution axiom schema* for L.
- (TL): $\vdash \mathrm{L}f \rightarrow f$, the *reflexivity axiom schema* for L. Unlike L, the axiom of reflexivity does not hold for [H]. It is important to remark that there are so far no connections between L and [H]. We force this connection by adding the following *link axiom schema*:
- (LI): $\vdash \neg(\mathrm{L}f \wedge H\neg f)$.

This very weak axiom is the base 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: $\neg(\mathrm{L}f \wedge \mathrm{H}\neg f) \Leftrightarrow \mathrm{L}f \rightarrow \neg\mathrm{H}\neg f \Leftrightarrow \mathrm{H}\neg f \rightarrow \neg\mathrm{L}f$, where the second formula means that *if we prove f, we cannot assume the hypothesis $\neg f$* and the third formula means that *if we assume the hypothesis $\neg f$, we cannot prove f*).

3.2 Semantics

The *Kripke semantics* [16] is 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 $\{w_1, \ldots, w_n\}$ 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 atom 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, $\mathrm{L}f$ 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. For the system K, R is any relation, while reflexivity axiom (TL) holds if and only if R is reflexive.

As shown in [30], $\mathcal{H}$ 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, is given by the extra constraint RL $\subseteq$ R[H] which corresponds to the link axiom. Proofs of completeness, correctness, and compactness for $\mathcal{H}$ are given in [30].

Note 1. The axiomatics of $\mathcal{H}$ was defined to give an alternative to Default Logic, using the minimum of axioms. With the full definition of $\mathcal{H}$, one could move to the system S4, by adding axiom (4) $\vdash \mathrm{L}f \rightarrow \mathrm{LL}f$. But this addition makes lose the notion of dynamics necessary to the representation of a BDS and thus of its underlying genetic network. Indeed, to represent the dynamics we consider here that $\mathrm{L}f$ represents an action (for example the production of a protein f), at a time step represented by a Krypke world w. All the accessible worlds from w are the following steps. Using (4), we obtain $\mathrm{LL}p$, and thus $\mathrm{L}p$ by (TL), in all these accessible worlds. Which, by induction thay means that p will be produced all the time. In a biological framework and for dynamical systems, that makes little sense.

3.3 Hypothesis theories and extensions

As defined above, $\mathcal{H}$ is a non-monotonic logic. In order to deal with the *revisable* character of usual informations, for example 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 $\mathcal{T} = \{\mathrm{HY}, \mathrm{F}\}$, where* HY *is a set of hypotheses and* F *is a set of formulas of $\mathcal{H}$.*
• *An* extension *E of $\mathcal{T}$ is a set $E = \mathrm{Th}(\mathrm{F} \cup \mathrm{HY}')$, such that HY' is a maximal subset of* HY *consistent with* F*.*
• *A* sub-extension *E of $\mathcal{T}$ is a set $E = \mathrm{Th}(\mathrm{F} \cup \mathrm{HY}')$, 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*:*

$$\forall \mathrm{H}f,\ \neg \mathrm{H}f \in E \implies \mathrm{L}\neg f \in E.$$

Thanks to the link axiom schema, we hence get: $\forall f,\ \mathrm{L}\neg f \in E \iff \neg \mathrm{H}f \in E$.
• *E is a* ghost extension *if it is an extension that satisfies: $\exists \mathrm{H}f,\ \neg \mathrm{H}f \in E$ and $\mathrm{L}\neg f \notin E$.*

Thus, an extension is obtained by adding one of the largest consistent sets of hypotheses to 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. Stable extensions correspond to the standard extensions of DL. Ghost extensions do not have any correspondence in DL nor in ASP. In [32, 30] it is proved that if F is consistent then $\mathcal{T} = \{\mathrm{HY}, \mathrm{F}\}$ has at least one extension and, that a default theory Δ can be translated into a hypothesis theory $\mathcal{T}(\Delta)$ such that the set of standard extensions of Δ is isomorphic to the set of stable extensions of $\mathcal{T}(\Delta)$.

The following definitions will help to characterize the stable configurations, cycles and the attractors of BDSs.

Definition 2. *Let E be an extension or a sub-extension:*
• *An $i \in V$ is* free *in E if $\mathrm{L}i \notin E$ and $\mathrm{L}\neg i \notin E$. It is* fixed *otherwise.*
• *The* degree of freedom *of E is the number of free atoms that compose it.*
• *The* mirror *of E, denoted by $\mathrm{mir}(E)$, is the set $\mathrm{Th}(\mathrm{F}(G) \cup \{\mathrm{H}\neg y_k\})$.*

Algorithm: There is not enough space to study the algorithms that computes the extensions. But, with the restricted definition of $\mathcal{H}$, used here there is no nesting of modalities (no $\mathrm{LL}f$, $\mathrm{HL}f$ ...), so all modal formulas are of the form $\mathrm{L}f$, where f is a PC formula. This restriction allows to translate $\mathcal{H}$ into PC by identifying any formula L with an atom of CP. We can then consider that the language of representation is finite, which opens the way to use SAT algorithms.

4 Representing genetic networks into $\mathcal{H}$

A *genetic network* represents interactions among genes or proteins in cell [2, 7, 8, 13, 15, 21]. In a modeling context, a protein is classically represented by an integer $i \in \{1, \ldots, n\}$. 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 $\{0, 1\}$.

Genetic networks can be studied with the formalism of BNs and their underlying BDSs, defined in the following section. Here, to introduce our representation 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. To lighten the notations, we will identify a numbered Boolean variable x_i directly with i.

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

We can then give the meanings of L and H in the context of genetic networks.

- i means that the protein i *is present in the cell and* $\neg i$ *that it is absent.*
- Li means that i *is produced by the cell (i is being activated) and* $\neg$Li *means that i is not produced (i is not being activated).*
- L$\neg i$ means that i *is destroyed by the cell (i is being inhibited) and* $\neg$L$\neg i$ *means that i is not destroyed (i is not being inhibited).*
- Hi (resp. $\neg$Hi) 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.*
- H$\neg i$ (resp. $\neg$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 Hi 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 H$\neg i$ gives the authorization to destroy i. It is important to note that Li and L$\neg i$ are actually actions (production or destruction of a protein). So there is a difference between L$\neg i$ which says that i is destroyed, and $\neg$Li which says that i is not produced, and hence is weaker. There is a similar distinction between H$\neg i$ and $\neg$Hi (and between Li and $\neg$L$\neg i$; and between Hi and $\neg$H$\neg i$).

Note 2. For a cell, the production/destruction occurs practically at a time t. This temporal notion could call for the use of temporal logics [3]. In our approach, using $\mathcal{H}$, it is not necessary to use a specific modality to represent time, which is implicitly included in the axiomatics of H, via the accessibility relation of Krypke semantics. As such, $\mathcal{H}$ is adapted to the representation of the dynamics of change.

Proposition below gives 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)]* $\mathrm{L}i \to i$ *and* $\mathrm{L}\neg i \to \neg i$ *(*i.e.*, if i is produced (resp. destroyed), then i is present (resp. absent)).*
- *[(2)]* $\neg(\mathrm{L}i \wedge \mathrm{H}\neg i)$ *and* $\neg(\mathrm{L}\neg i \wedge \mathrm{H}i)$ *(It is impossible to produce (resp. destroy) i and to give the permission to destroy (resp. produce) i it at the same time).*
- *[(3)]* $\neg(\mathrm{L}i \wedge \mathrm{L}\neg i)$ *(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)* $Li \to i$ and $L\neg i \to \neg i$ are two instances of (1) from which we derive $\neg(\mathrm{L}i \wedge \mathrm{L}\neg i)$.

5 Boolean Dynamical Systems

A finite BDS describes the evolution of the interactions in a BN of a set $V = \{1, ..., n\}$ of n entities numbered from 1 to n, over discrete time. A *configuration* $x = (x_1, ..., x_n)$ 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), called the *configuration space*, is denoted by $X = \{0, 1\}^n$. A *dynamic* of such a network is modeled *via* both a function f, called the *global transition function*, and an *updating mode* μ that defines how the elements of V are updated over time. More formally, $f : X \to X$ is such that $x = (x_1, ..., x_n) \mapsto f(x) = (f_1(x), ..., f_n(x))$, where each function $f_i : X \to \{0, 1\}$ is *a local transition function* that gives the evolution of i over time.

There exists an uncountable number of updating modes among which the *parallel* and the *fully asynchronous* ones remain the most used [14, 34]. The *parallel* updating mode is such that all the entities of the network are updated at each time step. Conversely, the *fully 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 fully asynchronous dynamics [23, 26] which we will abreviate by asynchronous dynamics in the sequel for the sake of simplicity.

5.1 Asynchronous transition graphs

Let $X = \{0, 1\}^n$ be a configuration space and $f : X \to X$ a function that defines a BN. The *asynchronous dynamics* of f is given by its asynchronous TG (ATG) $\mathscr{G}(f) = (X, T(f))$, a digraph whose vertex set is the configuration space and arc set is the set of effective asynchronous transitions such that:

$T(f) = \{(x, y) \in X^2 \mid \exists i \in V, x = (x_1, ..., x_i, ..., x_n),$
$y = (x_1, ..., x_{i-1}, f_i(x), x_{i+1}, ..., x_n), x \neq y\}.$

Therefore, if $(x, y) \in T(f)$, x and y differ exactly by one element; the transition is *unitary.*

An *orbit* in $\mathscr{G}(f)$ 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 $r \geqslant 2$ 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 $\mathscr{G}(f)$, *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 $\mathscr{G}(f)$, $\forall t < r$, 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 non trivial 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 or a stable oscillation, 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 positive and negative circuits of size 3:** Consider $V = \{1, 2, 3\}$, $x = \{0, 1\}^3$ and two functions/BNs 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, $\mathscr{G}(f)$ and $\mathscr{G}(g)$, pictured in Figure 1. A transition corresponds to one arrow in the picture. We notice that, the transition graph being asynchronous, for each arrow (x, y) then x differs from y by a single component.

There are up to 3 transitions leaving each configuration. Here, $\mathscr{G}(f)$ has two miror (symetric) stable configurations, $(\neg 1, 2, \neg 3)$ and $(1, \neg 2, 3)$ while all the other configurations belong to an unstable cycle. $\mathscr{G}(g)$ 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 $\mathscr{G}(f)$. We will see in Section 6 that the two stable configurations of $\mathscr{G}(f)$ correspond to two stable extensions of $\mathcal{H}$, and that the stable cycle of $\mathscr{G}(g)$ corresponds to a set of 6 ghost extensions of degree 1.

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

Example 3. Consider function/BN $k(x_1, x_2) = (x_2, x_1 \wedge \neg x_1 \wedge x_2)$, pictured in Figure 2. This ATG has a stable state $\{\neg 1, \neg 2\}$ and no cycles.

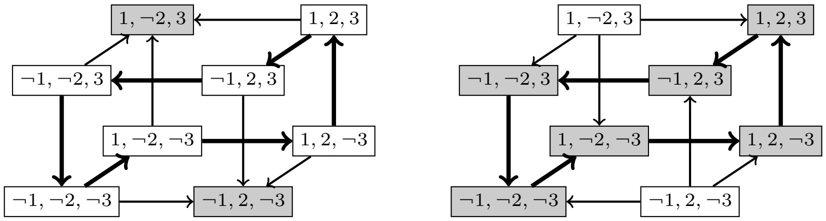

Fig. 1. ATG's of function *(left)* f and *(right)* g presented in Example 1.

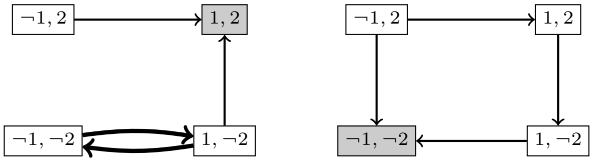

Fig. 2. *(left)* ATG of function h presented in Example 2 ; *(right)*, ATG of function k presented in Example 3.

5.2 General fundamental results and their biological direct applications

By considering that BNs and their associated BDSs are good candidates for qualitatively modeling genetic networks (since established by the seminal papers [14, 34]), 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 toward which its underlying BDS evolves. A classical example of direct biological applications is the immunity control in bacteriophage λ, for which both lytic and lysogenic cycles of λ have been modeled in [33]. Another more tricky applications of BDSs in molecular systems biology concerns the floral morphogenesis of the plant *Arabidopsis thaliana* [20, 21]. Its dynamical behavior admits notably four stable configurations that correspond to the genetic expression patterns of the floral tissues, sepals, petal, stamens and carpels.

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 *interactions graphs* and *transition graphs* 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 [27, 28]. Moreover, in [35], 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 proven to be true [23–26]. Furthermore, in [23], the authors showed that an asynchronous positive (resp. negative) circuit of size n admits two attractors (resp. one attractor), namely two stable configurations x and its dual $\overline{x}$ (resp. a stable cycle of length $2n$). In [31], we obtained these results *via* the translation of BDSs into $\mathcal{H}$.

6 Representing BDS into $\mathcal{H}$

In [31], 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. But this previous approach left formulas of the type $(\mathrm{H}i \wedge \mathrm{H}j) \rightarrow \mathrm{L}k$ out of reach. Such formulas are essential, for example, for representing the notion of *binding* in genetic networks. In the sequel, we extend this translation to any asynchronous BDS. This translation does not use nesting of modalities and SAT algorithms can be used.

6.1 Syntax representation of BDS

Remind that an asynchronous BDS is characterized by a function/ATG $f : X \rightarrow X$ such that $x = (x_1, \ldots, x_n) \mapsto f(x) = (f_1(x), \ldots, f_n(x))$, where each function $f_i : X \rightarrow \{0, 1\}$ is a local transition function. Also, remind that we consider that each x_i is an atom 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: $TR(f_i) = \{\mathrm{H}f_i(x) \rightarrow \mathrm{L}i$ and $\mathrm{H}\neg f_i(x) \rightarrow \mathrm{L}\neg i\}$.*
- *The translation of $f : X \rightarrow X$ of a BDS in $\mathcal{H}$ is the union of translations $TR(f_i)$ for all $i \in \{1, ..., n\}$ such that $TR(f) = \bigcup_{i=1}^{n} TR(f_i(x))$.*

Example 4. Consider $V = \{1, 2, 3\}$, $X = \{0, 1\}^3$, and the function f of Example 1, defined as $f(x) = (f_1(x), f_2(x), f_3(x)) = (\neg x_2, \neg x_3, x_1)$. The functions f_1, f_2 and f_3 are translated into $\mathcal{H}$ by;
$TR(f_1)$=$\{\mathrm{H2} \rightarrow \mathrm{L}\neg 1, \mathrm{H}\neg 2 \rightarrow \mathrm{L1}\}$, $TR(f_2)$=$\{\mathrm{H3} \rightarrow \mathrm{L}\neg 2, \mathrm{H}\neg 3 \rightarrow \mathrm{L2}\}$, $TR(f_3)$= $\{\mathrm{H1} \rightarrow \mathrm{L3}, \mathrm{H}\neg 1 \neg \mathrm{L3}\}$.

Therefore we obtain the following global translation:
$TR(f) = \{\mathrm{H2} \rightarrow \mathrm{L}\neg 1, \mathrm{H}\neg 2 \rightarrow \mathrm{L1}, \mathrm{H3} \rightarrow \mathrm{L}\neg 2, \mathrm{H}\neg 3 \rightarrow \mathrm{L2}, \mathrm{H1} \rightarrow \mathrm{L3}, \mathrm{H}\neg 1 \rightarrow \neg \mathrm{L3}\}$ that admits two stable extensions $E1 = Th(TR(f) \cup \{\mathrm{H1}, \mathrm{H}\neg 2, \mathrm{H3})\})$ and $E2 = Th(TR(f) \cup \{\mathrm{H}\neg 1, \mathrm{H2}, \mathrm{H}\neg 3)\})$ [5].

[5] This is shown by attempting to add to $\mathrm{F}(G(f))$ each subset of $\mathrm{HY}(G(f))$ and keeping only those among them that are the maximals ones consistent with $\mathrm{F}(G(f))$. This can be done using a SAT solver.

When developing these extensions, we see that they are equivalent to their simplified forms:

- $E1 = \{\mathrm{H}\neg 1, \mathrm{H}2, \mathrm{H}\neg 3, \mathrm{L}1, \mathrm{L}\neg 2, \mathrm{L}3, \neg\mathrm{H}1, \neg\mathrm{H}\neg 2, \neg\mathrm{H}3, \neg\mathrm{L}\neg 1, \neg\mathrm{L}2, \neg\mathrm{L}\neg 3\}$
- $E2 = \{\mathrm{H}1, \mathrm{H}\neg 2, \mathrm{H}3, \mathrm{L}\neg 1, \mathrm{L}2, \mathrm{L}\neg 3, \neg\mathrm{H}\neg 1, \neg\mathrm{H}2, \neg\mathrm{H}\neg 3, \neg\mathrm{L}1, \neg\mathrm{L}\neg 2, \neg\mathrm{L}3\}$.

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, $E1 = \{\mathrm{L}1, \mathrm{L}\neg 2, \mathrm{L}3\}$ and $E2 = \{\mathrm{L}\neg 1, \mathrm{L}2, \mathrm{L}\neg 3\}$. We can check that $E1$ and $E2$ are stable extensions (because for all i, $\neg\mathrm{H}i \in E1$ (resp. $E2$) $\implies$ $L\neg i \in E1$ (resp. $E2$) and that $E2$ is the miror of $E1$.

Example 5. Consider function k, presented in Example 3, 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 right. Function k_1 is translated into $\mathcal{H}$ by the couple $TR(k_1) = \{\mathrm{H}2 \rightarrow \mathrm{L}\neg 1, \mathrm{H}\neg 2 \rightarrow \mathrm{L}1\}$ and function k_2 is translated by $TR(k_2) = \{\mathrm{H}(1 \wedge \neg 1 \wedge 2) \rightarrow \mathrm{L}2), \mathrm{H}\neg(1 \wedge \neg 1 \wedge 2) \rightarrow \mathrm{L}\neg 2\}$. 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: $TR(k) = \{\mathrm{H}2 \rightarrow \mathrm{L}\neg 1, \mathrm{H}\neg 2 \rightarrow \mathrm{L}1, \mathrm{H}(1 \wedge \neg 1 \wedge 2) \rightarrow \mathrm{L}2), \mathrm{H}(\neg 1 \vee 1 \vee \neg 2) \rightarrow \mathrm{L}\neg 2)\}$, which admits thee extensions:

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

Function k may appear naive, because $x_1 \wedge \neg x_1 \wedge x_2 = \bot$, which gives an equivalent translation $TR(h) = \{\mathrm{H}2 \rightarrow \mathrm{L}\neg 1, \mathrm{H}\neg 2 \rightarrow \mathrm{L}1, \mathrm{H}\bot \rightarrow \mathrm{L}2, \mathrm{H}\top \rightarrow \mathrm{L}\neg 2\}$. 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 implicitly make the expected simplifications.

6.2 Semantic representation of ATGs into $\mathcal{H}$

This section gives a morphism between ATGs and Kripke models for the modal system T, which allows us to exhibit a morphism from hypothesis theories to ATGs. It uses *Kripke semantics* [16] presented in Section 3.2. In order to obtain these morphisms, we give and increased version of ATGs.

Definition 4. *Let $V = \{1, \ldots, n\}$ be a set of entities, $X = \{0,1\}^n$ be a configuration space, $f : X \rightarrow X$ be a function with its associated ATG $\mathscr{G}(f) = (X, T(f))$. Remind that $T(f)$ is a set of edges corresponding to transitions. We now look at an increased version of $\mathscr{G}(f)$, namely $\mathscr{G}^\star(f) = (X, T \cup \circlearrowleft)$ where $\circlearrowleft$ denotes the reflexivity such that (x, x) is a transition of $\mathscr{G}^\star(f)$ for all $x \in \{0,1\}^n$.*

We can consider that $\mathscr{G}^\star(f)$ is a Kripke structure whose universe is X and whose accessibility relation is $R = T(f) \cup \circlearrowleft$. If we consider that any configuration $x \in X$ is a world, we get a Kripke model with R as accessibility relation. As R is reflexive, it has the properties of system T. Therefore, we get an isomorphism between reflexive Kripke models and increased ATGs, from which it is trivial to obtain the related ATGs.

As the ATG $\mathscr{G}(f)$ is asynchronous, the accessibility relation R is such that, if $w_j \neq w_k$, then $w_j \, R \, w_k$ if and only if, w_k is reachable from w_j and differs from

w_j by one and only one proposition. Under these conditions, the ATG of any BDS is a canonical Kripke structure. Given such a framework, for any world w_k and any entity x, $\mathrm{L}x = \top$ if and only if $x = \top$ for every w_k reachable from w_j.

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 5. *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 $\mathcal{T} = \{\mathrm{HY}, \mathrm{F}\}$ 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 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.

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

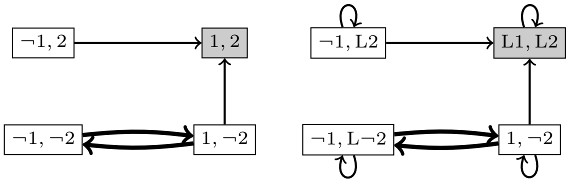

Fig. 3. *(left)* The ATG of h, *(rigth)* the corresponding Kripke model.

6.3 Results

The existence of a morphism from hypothesis theories to ATGs, allows to prove these theoretical results. Therefore is a strong link between configurations and extension sets.

The generalization of the notion of degree of freedom of extensions allows to configure ATGs. The *degree of a configuration* x, is the number of arcs coming out of x. By previous construction of the Kripke models associated with ATGs, it is obvious that, if w is the world associated with x, then x and w have the same degree.

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 $\{i_1, .., i_k\}$ atoms, free in E. For every $i \in \{i_1, \dots, i_k\}$, we have both $\neg \mathrm{L}i \in E$ and $\neg \mathrm{L}\neg i \in E$. Two cases are possible, either $i \in E$ or $\neg i \in E$. If $i \in E$, since $\neg \mathrm{L}i \in E$, 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 $\neg \mathrm{L}\neg i \in E$, 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 ATG all these accessible worlds are distinct. Hence there are k distinct worlds reachable from w.

Theorem 1. *Let $\mathscr{G}(f)$ 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 $\mathscr{G}(f)$, then there exists an extension E of degree 0, issued from $TR(f)$, that contains $\{\mathrm{L}x_1, \dots, \mathrm{L}x_n\}$.

[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 proven separately:

[1.] If x is a stable configuration of $\mathscr{G}(f)$, 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 word accessible from w is w, that is, for any $i \in w$ (resp. $\neg i \in w$), $\mathrm{L}i \in w$ (resp. $\mathrm{L}\neg i \in w$). 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 $\mathscr{G}(f)$.

Theorem 2. *Let $\mathscr{G}(f)$ be the ATG of function f and $TR(f)$ be its associated hypothesis theory. Every stable cycle C of $\mathscr{G}(f)$ 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 $\mathscr{G}(f)$, 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 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 interaction graphs [31]. They correspond to the results given in [23]. 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.

7 Conclusion

In [31], 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 captures some of their essential behavioral capacities, 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.

References

1. Akman, O.E., Watterson, S., Parton, A., Binns, N., Millar, A.J., Ghazal, P.: Digital clocks: simple Boolean models can quantitatively describe circadian systems. Journal of The Royal Society Interface **9**, 2365–2382 (2012)
2. Aracena, J., González, M., Zuñiga, A., Mendez, M.A., Cambiazo, V.: Regulatory network for cell shape changes during Drosophila ventral furrow formation. Journal of Theoretical Biology **239**, 49–62 (2006)
3. Besnard, P., Siegel, P.: Supposition-Based Logic for Automated Nonmontonic Reasoning. In: Proceedings of CADE'88. LNCS, vol. 310, pp. 592–601. Springer (1988)
4. Chellas, B.: Modal logic: an introduction. Cambridge University Press (1980)
5. Davidich, M.I., Bornholdt, S.: Boolean network model predicts cell cycle sequence of fission yeast. PLoS One **3**, e1672 (2008)
6. Delgrande, J.P., Schaub, T.: Expressing default logic variants in default logic. Journal of Logic and Computation **15**, 593–621 (2005)
7. Demongeot, J., Elena, A., Noual, M., Sené, S., Thuderoz, F.: "Immunetworks", intersecting circuits and dynamics. Journal of Theoretical Biology **280**, 19–33 (2011)
8. Demongeot, J., Goles, E., Morvan, M., Noual, M., Sené, S.: Attraction basins as gauges of the robustness against boundary conditions in biological complex systems. PLoS One **5**, e11793 (2010)
9. Demongeot, J., Noual, M., Sené, S.: Combinatorics of Boolean automata circuits dynamics. Discrete Applied Mathematics **160**, 398–415 (2012)
10. Dergacheva, O., Griffioen, K.J., Neff, R.A., Mendelowitz, D.: Respiratory modulation of premotor cardiac vagal neurons in the brainstem. Respiratory Physiology & Neurobiology **174**, 102–110 (2010)
11. Doncescu, A., Siegel, P.: DNA double-strand break-based nonmonotonic logic. In: Emerging Trends in Computational Biology, Bioinformatics, and Systems Biology, pp. 409–427. Elsevier (2015)
12. Doncescu, A., Siegel, P., Le, T.: Representation and efficient algorithms for the study of cell signaling pathways. In: Proceedings of ICAI'14. pp. 504–510. IEEE Computer Society (2014)
13. Fauré, A., Naldi, A., Chaouyia, C., Thieffry, D.: Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. Bioinformatics **22**, e124–e131 (2006)
14. Kauffman, S.A.: Metabolic stability and epigenesis in randomly constructed nets. Journal of Theoretical Biology **22**, 437–467 (1969)

15. Kauffman, S.A., Peterson, C., Samuelsson, B., Troein, C.: Random Boolean network models and the yeast transcriptional network. PNAS **100**, 14796–14799 (2003)
16. Kripke, S.A.: Semantical analysis of modal logic I Normal modal propositional calculi. Mathematical Logic Quarterly **9**, 67–96 (1963)
17. Li, F., Long, T., Lu, Y., Ouyang, Q., Tang, C.: The yeast cell-cycle network is robustly designed. PNAS **101**, 4781–4786 (2004)
18. Lifschitz, V.: Action languages, answer sets, and planning. In: The Logic Programming Paradigm: A 25-Year Perspective, pp. 357–373. Springer (1999)
19. Melliti, T., Noual, M., Regnault, D., Sené, S., Sobieraj, J.: Asynchronous dynamics of Boolean automata double-cycles. In: Proceedings of UCNC'15. LNCS, vol. 9252, pp. 250–262. Springer (2015)
20. Mendoza, L., Alvarez-Buylla, E.R.: Dynamics of the genetic regulatory network for *Arabidopsis thaliana* flower morphogenesis. Journal of Theoretical Biology **193**, 307–319 (1998)
21. Mendoza, L., Thieffry, D., Alvarez-Buylla, E.R.: Genetic control of flower morphogenesis in *Arabidopsis thaliana*. Bioinformatics **15**, 593–606 (1999)
22. Reiter, R.: A logic for default reasoning. Artificial Intelligence **13**, 81–132 (1980)
23. Remy, E., Mossé, B., Chaouyia, C., Thieffry, D.: A description of dynamical graphs associated to elementary regulatory circuits. Bioinformatics **19**, ii172–ii178 (2003)
24. Remy, E., Ruet, P., Thieffry, D.: Graphic requirement for multistability and attractive cycles in a Boolean dynamical framework. Advances in Applied Mathematics **41**, 335–350 (2008)
25. Richard, A.: Negative circuits and sustained oscillations in asynchronous automata networks. Advances in Applied Mathematics **44**, 378–392 (2010)
26. Richard, A., Comet, J.P.: Necessary conditions for multistationarity in discrete dynamical systems. Discrete Applied Mathematics **155**, 2403–2413 (2007)
27. Robert, F.: Itérations sur des ensembles finis et automates cellulaires contractants. Linear Algebra and its Applications **29**, 393–412 (1980)
28. Robert, F.: Discrete Iterations: A Metric Study. Springer (1986)
29. Roenneberg, T., Merrow, M.: The network of time: understanding the molecular circadian system. Current Biology **13**, R198–R207 (2003)
30. Schwind, C., Siegel, P.: A modal logic for hypothesis theory. Fundamenta Informaticae **21**, 89–101 (1994)
31. Siegel, P., Doncescu, A., Risch, V., Sené, S.: Towards a Boolean dynamical system representation into a nonmonotonic modal logic. In: Proceedings of NMR'18. pp. 53–62 (2018)
32. Siegel, P., Schwind, C.: Modal logic based theory for non-monotonic reasonning. Journal of Applied Non-classical Logic **3**, 73–92 (1993)
33. Thieffry, D., Thomas, R.: Dynamical behaviour of biological regulatory networks – II. Immunity control in bacteriophage Lambda. Bulletin of Mathematical Biology **57**, 277–297 (1995)
34. Thomas, R.: Boolean formalization of genetic control circuits. Journal of Theoretical Biology **42**, 563–585 (1973)
35. 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, pp. 180–193. Springer (1981)