Towards a boolean dynamical system representation in a monmonotonic modal logic

Pierre Siegel^1, **Andrei Doncescu**^{2}, **Vincent Risch**^{1}, **Sylvain Sené**^{1}
^{1} Aix-Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
^{2} Université Paul Sabatier CNRS, LAAS, Toulouse, France
{pierre.siegel,vincent.risch,sylvain.sene}@univ-amu.fr, andrei.doncescu@laas.fr

Abstract

Boolean dynamical systems (BDSs) represent the evolution of interactions inside a finite network of entities taking Boolean states over discrete time. These networks are classically used to model interactions of biological networks. In this context, a genetic network can be represented by both a Transition Graph (TG) and an Interaction Graph (IG). The precise relationship between IG and TG has been studied for many years in dynamical systems theory while still an open question. The global purpose of this article is to further study this relationship *via* a logical representation of BDSs into a nonmonotonic modal logic called Hypothesis Logic ($\mathcal{H}$). While the *dynamics* of a BDS are characterized by a function f, an important part of the studies focused on the analysis of both stable configurations (i.e. fixed points of f), and stable/unstable cycles of f. For the representation of some genetic networks with no negative feedback circuits, results were previously obtained with some well known nonmonotonic formalisms. So far however, BDSs representation by most of these formalisms does not permit to capture cyclic dynamical behaviors. Notably, the equivalent of a negative circuit has no extension in default logic (DL). This is embarrassing because these cycles may represent real interactions in living organisms like the cell cycle. This possible lack of extensions in DL was studied in $\mathcal{H}$, for which theories always have extensions while some of these, called *ghost extensions*, are actually not extensions of the corresponding theories in DL. This paper addresses to the question of a first representation of the dynamics of BDSs with $\mathcal{H}$, and ghost extensions appear to be a powerful tool in this respect. As we are especially concerned with cycles, it provides us with hints of simple algorithms for computing exhaustively both stable/unstable cycles and fixed points: distinguishing between stable/unstable as well as enumerating all the solutions in practice would be a major advance that would lead to apprehend better inner fundamental aspects in biology.

1 Introduction

Biological networks are representations of the bioprocesses on three levels of interactions into the biological cell: genomic, protein and metabolic level. The techniques which are the most used in Network Modeling are boolean networks, bayesian belief network and metabolic network modeling methods. The boolean network are well adapted for Gene Regulatory Networks. At this level, it is considered the genes are on-off switches which do not act independently. Two genes are connected if the expression of one gene modulates the expression of another one by activation or inhibition. It is interesting to remark some genes control the response of the cell to changes in the environment by regulating other genes.

From a logical point of view, a biological system can be considered as a set of interacting elements changing along discrete time. Genetic networks are specific biological systems that represent how the proteins/genes in a cell interact for the survival, reproduction, or death of this cell. These interactions can be studied in the context of automata networks and Boolean dynamical systems (BDSs) as a set of entities taking Boolean states. Founding theorems (Demongeot, Noual, and Sené 2012; Melliti et al. 2015; 2013; Remy et al. 2003; Remy, Ruet, and Thieffry 2008; Richard 2010; Richard and Comet 2007; Robert 1986; Thomas 1981) have focused on feedback circuits (simply called circuits hereafter).

A genetic network can be represented by both a *transition graph* (TG) and an *interaction graph* (IG). The relationship between TG and IG has been studied for many years in dynamical systems theory, but remains an important open question. This paper adresses this question with help of a representation of BDSs *via* a nonmonotonic modal logic called Hypothesis Logic ($\mathcal{H}$): our logic-based approach seems to be a step toward a global clarification of this relationship. Preliminary results were already given in (Siegel et al. 2017).

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. Moreover, what we learn arises largely from long and expensive experiments. Hence, we know only a small part of the interactions while this knowledge can be revisable, uncertain, contradictory and even false. Eventually, 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. In particular it is possible to use default logic (DL) (Reiter 1980) and answer set programming formalism (ASP) (Lifschitz 1999).

The *dynamics* of a BDS are characterized by a function f and an updating mode. An important part of the studies done on BDSs focused on the analysis of both stable configurations (i.e. fixed points of f), and stable/unstable cycles of f. For the representation of some genetic networks with no negative feedback circuits, results have been obtained with DL (Doncescu, Siegel, and Le 2014),(Doncescu and Siegel 2015). Representation of a BDS by whatever DL, ASP or other nonmonotonic formalisms, enables to find fixed points. But, these representations are not suitable for finding cycles. Notably, the equivalent of a negative circuit has no extensions in default logic. This is embarrassing because these cycles may represent real interactions in living organisms like the cell cycle (Davidich and Bornholdt 2008; Li et al. 2004), or the circadian cycle (Akman et al. 2012; Roenneberg and Merrow 2003). This possible lack of extensions in default logic has been fully studied in the context of *hypothesis logic* ($\mathcal{H}$) (Schwind and Siegel 1994; Siegel and Schwind 1993). In this logic, theories always have extensions while some of these, called *ghost extensions*, are actually not extensions of the corresponding theories in default logic. Moreover, very simple and efficient algorithms, used for solving SAT problems, can be applied to the computation of extensions, fixed points and cycles.

The representation of the dynamics of BDSs in $\mathcal{H}$ aims at making possible to discriminate between stable configurations (fixed points), limit cycles and unstable cycles. We introduce representations for both Interaction Graphs and Asynchronous Transition Graphs in $\mathcal{H}$, which allows us to exhibit new formal results. Ghost extensions play here a key role. This approach provides us with hints of simple algorithms for distinguishing between stable/unstable as well as enumerating all the solutions in practice. This would be a major advance regarding better inner fundamental aspects in biology.

In what follows, Section 2 presents basic definitions for BDSs and Section 3 reminds the basics of nonmonotonic and hypothesis logic. Section 4 gives a representation of IGs into $\mathcal{H}$ and prove some properties related to this representation. Section 5 studies relationships between ATGs of a BDS and $\mathcal{H}$: Theorem 3 states that there exists an isomorphism between stable configurations and stable extensions, and Theorem 4 states that every negative feedback circuit admits a set of ghost extensions whose semantics is analogous to BDS dynamical cycles.

2 Finite Boolean dynamical systems

A finite discrete dynamical system (DDS) describes the evolution of the interactions in a network of n entities numbered from 1 to n, over discrete time. This evolution is the *dynamics* of the system. An example of such a system is the representation of genetic networks, namely networks representing interactions between the genes or the proteins of a cell (Aracena et al. 2006; Demongeot et al. 2011; 2010; Fauré et al. 2006; Kauffman et al. 2003; Mendoza, Thieffry, and Alvarez-Buylla 1999). In the context of genetic networks, an entity $i \in \{1, ..., n\}$ depicts a protein whose concentration is denoted by x_i so that there is only a finite number of possible concentrations. In such networks, given a protein i, a set of interactions for proteins on i gives the conditions for this set to increase or decrease the concentration of i.

Regarding *Boolean dynamical systems* (BDSs) studied in this paper, the concentrations x_i are in $\{0, 1\}$. In this case, $x_i = 1$ denotes the presence of i and $x_i = 0$ its absence. Yet $x_i = 1$ (resp. $x_i = 0$) is the $activation$ (resp. $inhibition$) or the $production$ (resp. $destruction$) of the protein.

Note that what is written $x_i = 0$ (resp. $x_i = 1$) in the context of BDS corresponds logically to $v(i) = 0$ (resp. $v(i) = 1$), where v is the standard valuation function of propositional logic. Hence, a protein i is nothing else than a propositional variable. Abusing the notations we authorize ourselves to write either $i, \neg i$ as well as $x_i, \neg x_i$ depending on the context. For instance, $x = (\neg x_1, x_2, \neg x_3)$ corresponds to $x = (\neg 1, 2, \neg 3)$ or even simply $(\overline{1}, 2, \overline{3})$.

Consider $V = \{1, ..., n\}$ a set of n entities. A *configuration* $x = (x_1, ..., x_n)$ of the system is an assignment of a truth value $x_i \in \{0, 1\}$ to each element i of V. The set of all configurations (Jacob and Monod 1978), also called the *space of configurations*, is denoted by $X = \{0, 1\}^n$. The *dynamics* of such a system is modeled via both a function f, called the *global transition function*, and an *updating mode* that defines how the elements of V are updated along 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 the state x_i over time. There exists an infinite number[1] of updating modes among which the *parallel* and the *asynchonous* ones remain the most used. 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 (Melliti et al. 2015; Remy, Ruet, and Thieffry 2008; Richard and Comet 2007).

2.1 Asynchronous transition graphs

Every dynamic being characterized by a function f and an updating mode, an important part of studies done on BDSs focused on the analysis of both stable configurations (i.e. fixed points of f), and stable/unstable cycles of f. Regarding the asynchronous case, this study is fulfilled via the notion of *Asynchronous Transition Graph* (ATG) associated with f.

Let $X = \{0, 1\}^n$ be the configuration space and consider a function $f : X \to X$. The *asynchronous dynamics* of f is given by its ATG $\mathscr{G}(f) = (X, T(f))$, the digraph whose vertex set is the configuration space and arc set is the set of transitions: $T(f) = \{(x, y) \in X^2 \mid x \neq y, x = (x_1, ..., x_i, ..., x_n), y = (x_1, ..., x_{i-1}, f_i(x), x_{i+1}, ..., x_n)\}$

If $(x, y) \in T(f)$, then the Hamming distance between x and y equals 1 (the transition is *unitary*). An *orbit* in $\mathscr{G}(f)$ is a sequence of configurations $(x^0, x^1, x^2, ...)$ 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 $(x^1, ..., x^r, x^1)$ with $r \geq 2$ whose configura-

[1] Infinite, because deterministic updating modes are basically defined as infinite sequences of subsets of nodes of the network.

tions $x^1, ..., x^r$ are all different. From this, we derive what is classically called an asynchronous *attractor* in dynamical systems, namely a terminal *strongly connected component* (SCC) of $\mathscr{G}(f)$, *i.e.* a SCC with no outward transitions. Among attractors, in the sequel, we will pay particular attention to stable configurations (fixed points) and 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, on $\mathscr{G}(f)$, $\forall t < r$, 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 SCC, this SCC is called an unstable cycle or oscillation depending on which it is cyclic or not.

An orbit that reaches a stable configuration stays there endlessly. Similarly, if it reaches a stable cycle, it adopts endlessly a stable oscillating behavior.

Example 1 (Boolean positive and negative circuit of size 3**)** Consider $V = \{1, 2, 3\}$, $X = \{0, 1\}^3$ and the two following 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 functions f and g, it is easy to derive their related ATGs, $\mathscr{G}(f)$ and $\mathscr{G}(g)$, pictured in Figure 1. For each arc (x, y) in $\mathscr{G}(f)$ and $\mathscr{G}(g)$, if $x \neq y$ then x differs from y by a single component. There are up to 3 transitions leaving each configuration. Here, $\mathscr{G}(f)$ has two stable configurations, $(\neg 1, 2, \neg 3)$ and $(1, \neg 2, 3)$ while all the other configurations belong to an unstable cycle pictured in bold. $\mathscr{G}(g)$ has a stable cycle pictured in bold. This cycle is stable because there is only one transition (corresponding to one arrow in the picture) leaving from each configuration, which is not the case for the unstable cycle of $\mathscr{G}(f)$.

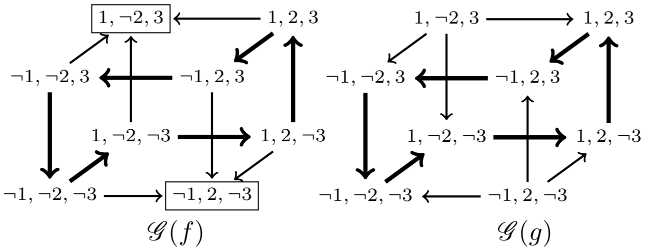
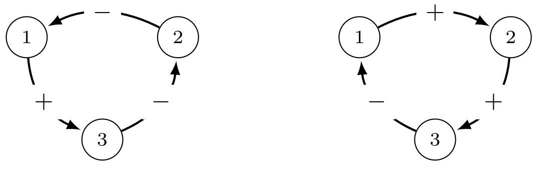


Figure 1: ATGs of functions f and g given in Example 1.



(a) *(b)*

Figure 2: *(a)* IG associated with ATG $\mathscr{G}(f)$ and *(b)* IG associated with ATG $\mathscr{G}(g)$ of BDSs defined in Example 1.

2.2 Interaction graphs and circuits

A TG is an excellent tool for studying the behavior of a function. But in practice biological data come from experiments that generally yield correlations among gene expressions. This information is classically modeled by interaction graphs, exponentially more compact and more "readable" than TGs. Contrary to TGs, these graphs only give static information about how entities act on each other.

The IG of a BDS of function f is induced by its local transition functions f_i. An important line of research on BDSs concerns what we can say about the TG of a BDS by knowing only its static specification, that is its function, and thus its IG.

An IG is a signed digraph $G = (V, I)$, where $V = \{1, ..., n\}$ is the set of vertices and $I \subseteq V \times S \times V$, with $S = \{-, +\}$. An arc $(i, +, j)$ (resp. $(i, -, j)) \in I$ is said to be *positive* (resp. *negative*). A *circuit* $C = \{(i_1, s_{(1,2)}, i_2), ..., (i_k, s_{(k,1)}, i_1)\}$ of size k in terms of graph theory is *elementary* if all i_s that compose it are distinct. A circuit is *positive* (resp. *negative*) when it contains an even (resp. an odd) number of negative arcs. From the BDS point of view, the presence of an arc (i, s, j) in an IG G means that the value of i affects that of j: we say that i regulates j.

Consider the toy example where j has only one incoming arc, from i. In this case, the effect of the regulation is very simple: if the arc is positive (resp. negative), the state of j will take the value (resp. the opposite value) of that of i after one update, such that $f_j(x) = x_i$ (resp. $f_j(x) = \neg x_i$). Notice that elementary circuits are regulated this way. For example Figure 2 pictures the IGs associated with the ATGs of the BDSs defined from f and g in Example 1.

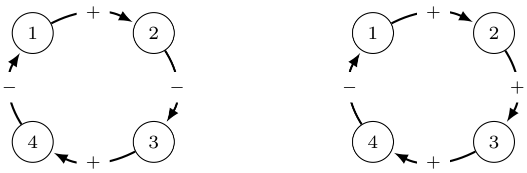
More generally, an IG $G = (V, I)$ represents the existence of the interactions involved between its entities in V. Specifying the nature of these interactions and the conditions under which they occur effectively leads to relate G to a BDS of function f, so that G becomes the IG of f and is then denoted by $G(f) = (V, I(f))$. This is done by assigning a local transition function f_i to every $i \in V$ so that $\forall j \in V, \exists x \in \{0, 1\}^n,\ f_i(x) \neq f_i(\overline{x}^j) \iff (j, s, i) \in I(f)$, where, given $x = (x_1, ..., x_n)$, $\overline{x}^j = (x_1, ..., x_{j-1}, \neg x_j, x_{j+1}, ..., x_n)$. We generalize this notation by $\overline{x} = (\neg x_1, ..., \neg x_n)$. Such a specification induces the minimality of $G(f)$ because each arc represents an effective interaction.

Note 1 Consider a BDS and its associated IG $G(f)$, such that arc (i, s, i) belongs to $I(f)$. If $s = +$ (resp $s = -$), this arc makes i tending 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 neighbors. In the case i admits no other in-neighbors, it is trivial that i endlessly maintains (resp. negate) its state if $s = +$ (resp. $s = -$).

Let us present now the asynchronous dynamical behaviors of a Boolean positive circuit and of a Boolean negative circuit of size 4 in Examples 2 and 3 below.

Example 2 (Boolean positive circuit of size 4**)** *Consider the BDS of function* $f(x_1, x_2, x_3, x_4) = (\neg x_4, x_1, \neg x_2, x_3)$. *Figure 3-a depicts the corresponding IG. This BDS admits two stable configurations,* $(1, 2\neg 3, \neg 4)$ *and* $(\neg 1, \neg 2, 3, 4)$, *and an unstable oscillation.*

Example 3 (Boolean negative circuit of size 4**)** *Consider the BDS of function* $g(x_1, x_2, x_3, x_4) = (\neg x_4, x_1, x_2, x_3)$*. Figure 3-b depicts the correponding IG. This BDS admits one stable cycle of length and one unstable cycle of length* 8



(a) *(b)*

Figure 3: *(a)* IG of Boolean positive circuit of size 4 (Example 2) and *(b)* negative circuit of size 4 (Example 3).

2.3 General central results

By considering that BDSs are good candidates for qualitatively modeling genetic networks (since established by the seminal papers of (Kauffman 1969) and (Thomas 1973)), the presence of several attractors in their dynamical behaviors allows to model the cellular specialization at the level of cells. 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. These works and the numerous other ones using BDSs (or, more generally, DDSs) highlighted the essential role of studies aiming at understanding the formal relations between IG and TG. 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 (Robert 1986). Moreover, in (Thomas 1981), Thomas conjectured that $G(f)$ of a 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 more complex one). These two conjectures have been proven to be true under the hypothesis of asynchronous updating mode (Remy et al. 2003; Remy, Ruet, and Thieffry 2008; Richard 2010; Richard and Comet 2007). Furthermore, notice that in (Remy et al. 2003), the authors showed that an asynchronous positive circuit of size n admits two attractors, namely two stable configurations x and $\overline{x}$, and that an asynchronous negative circuit admits only one attractor, namely a stable cycle of length $2n$.

Note 2 *While in the following we use positive and negative feedback circuits because of their central role in dynamic, our definitions and results hold for the full general framework. For instance the example proposed in Figure 4 is tractable inside our framework. Due to lack of space we do not treat here.*

3 Nonmonotonic, default, hypothesis logics

Representing IGs with a logical formalism seems natural because the way an arc (i, s, j) is interpreted suggests a close relation with what is called *material implication* in logic. 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 nonmonotonic formalisms, among which default logic or ASPs (a more tractable restriction of default logic). *Default logic* (DL) (Reiter 1980) concerns standard formulas 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/consistent 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 nonmonotonic because adding here a new information may invalidate previously triggered defaults.

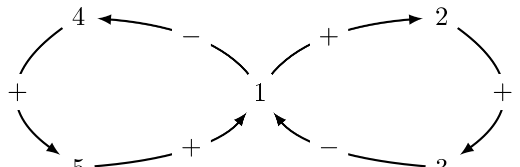


Figure 4: .

There is clearly a connection between default extensions and stable configurations of a BDS (Doncescu, Siegel, and Le 2014; Doncescu and Siegel 2015). Also, recent works have studied the connection between ASP stable models and stable configurations. A drawback is that these connections are limited to the representation of stable configurations only (included odd circuits)while it tells nothing about unstable cycles. The problem arises from the way to capture all the dynamics.

Another drawback, actually linked to the first one, is that in DL, some theories may simply have no extensions at all, thus depicting a form of deep inconsistency which renders computation more difficult. By definition, DL only computes stable extensions. Such a type of extension is limited since it appears too cheap to handle with more than stability while we expect to capture also unstable cycles, e.g. the simple default theory ($d = \frac{:A}{\neg B}, \frac{:B}{\neg C}, \frac{:C}{\neg B}$), linked to the representation of a negative circuit, has no stable extension.

This drawback is overcome by justified DL (Lukaszewicz 1988) as well as by hypothesis logic. While justified DL exclusively consists in a reformulation of the conditions by which extensions are obtained, hypothesis logic is both a reformulation and a generalization of DL inside the framework of a normal bimodal logic. It actually generalizes justified (and hence classical) DL in the following way: while stable extensions are a special case of justified extensions, justified extensions are in turn a special case of the extensions obtained by hypothesis logic. Besides, the specific way hypothesis logic answers the question of the lack of extensions in DL sheds a new light on the representation of the dynamics in BDSs. In the context of a *theoretical* study concerning BDSs, we will find in Logic the formal criterias that help delimiting the "good" properties of a *langage* devoted to such a study. This is indeed the role of Logic to state representation theorems between a langage and the objects this langage concerns. We expect hypothesis logic to be a right candidate as such a langage because it overtakes some of the limitations of DL (and hence ASP also).

3.1 Hypothesis Logic

Hypothesis logic $\mathcal{H}$ is a bi-modal logic[2] with two modal operators L and [H]. If f is a formula, the intuitive meaning of $\mathrm{L}f$ is f *is proved/stated*. The dual H of [H] is defined as $\mathrm{H}f = \neg[\mathrm{H}]\neg f$. It was introduced in (Schwind and Siegel 1994; Siegel and Schwind 1993). The intuitive meaning of $\mathrm{H}f$ is f *is a hypothesis*, and hence $[\mathrm{H}]f$ means $\neg f$ *is not a hypothesis*.

A default $\frac{A:B}{C}$ is interpreted in $\mathcal{H}$ by the modal formula $\mathrm{L}A \wedge \mathrm{H}B \to \mathrm{L}C$ whose intuitive meaning is: If A is stated and B is a valid hypothesis then C is stated. The language of $\mathcal{H}$, denoted by $\mathscr{L}(\mathcal{H})$, is defined by the following rules:

- Any formula of first-order logic is in $\mathscr{L}(\mathcal{H})$.
- If f and g are in $\mathscr{L}(\mathcal{H})$ then formulas $\neg f$, $(f \wedge g)$, $(f \vee g)$, $(f \to g)$, $\mathrm{L}f$, $[\mathrm{H}]f$, $\mathrm{H}f$ are too.

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]): $\vdash f \implies \vdash [\mathrm{H}]f$, the *necessitation rule* for [H].
- (NL): $\vdash f \implies \vdash \mathrm{L}f$, the *necessitation rule* for L.
- (K[H]): $\vdash [\mathrm{H}](f \to g)) \to ([\mathrm{H}]f \to [\mathrm{H}]g)$, the *distribution axiom schema* for [H].
- (K[L]): $\vdash \mathrm{L}(f \to g) \to (\mathrm{L}f \to \mathrm{L}g)$, the *distribution axiom schema* for L.
- (TL): $\vdash \mathrm{L}f \to f$, the *reflexivity axiom schema* for T.

Unlike L, [H] has no reflexivity axiom schema and, as it stands, there is so far no connections between L and [H]. We make this connection by adding the following *link axiom schema*:

$$(\mathrm{LI}) : \vdash \neg(\mathrm{L}f \wedge H\neg f).$$

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: $\neg(\mathrm{L}f \wedge \mathrm{H}\neg f) \iff \mathrm{L}f \to \neg\mathrm{H}\neg f \iff \mathrm{H}\neg f \to \neg\mathrm{L}f$. The first one means that *if we prove f, we cannot assume the hypothesis $\neg f$*, the second that *if we assume the hypothesis $\neg f$, we cannot prove f.*

3.2 Hypothesis theories and extensions

$\mathcal{H}$ is a monotonic logic. In order to catch nonmonotonicity, a notion of extension is added similarly to default logic. However, contrary to the latter, two kinds of extensions are considered here, namely stable extensions and ghost extensions. More formally :

Definition 1 *Let $\mathcal{H}$ be the hypothesis logic:*

- *A* hypothesis theory *is a pair $\mathcal{T} = \{\mathrm{HY}, \mathrm{F}\}$, 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 $E = \mathrm{Th}(\mathrm{F} \cup \mathrm{HY}')$, such that* HY' *is a maximal subset of* HY *consistent with* F.
- *E is a* stable extension *if it satisfies the* coherence property*: $\forall \mathrm{H}f,\ \neg\mathrm{H}f \in E \implies \mathrm{L}\neg f \in E$. Hence, given the link axiom schema, we get: $\forall f, \mathrm{L}\neg f \in E \Leftrightarrow \neg\mathrm{H}f \in E$*
- *E is a* ghost extension *otherwise,* i.e. *if it satisfies: $\exists \mathrm{H}f,\ \neg\mathrm{H}f \in E$ and $\mathrm{L}\neg f \notin E$*

Hence for a ghost extension, we only get: $\forall f, \mathrm{L}\neg f \in E \Rightarrow \neg\mathrm{H}f \in E$

In other words, ghost extensions are "pre"–stable extensions.

Theorems 1 and 2 below give fundamental properties of $\mathcal{H}$. Their proof are given in (Siegel and Schwind 1993; Schwind and Siegel 1994).

Theorem 1 *If* F *is consistent then $\mathcal{T} = \{\mathrm{HY}, \mathrm{F}\}$ has at least one extension.*

Theorem 2 *Let $\Delta = \{D, W\}$ be an arbitrary default theory. Δ can be translated into a hypothesis theory $\mathcal{T}(\Delta)$ such that:*

1. W is consistent, $\mathcal{T}(\Delta)$ admits at least one extension;

2. The set of standard extensions of Δ is isomorphic to the set of stable extensions of $\mathcal{T}(\Delta)$.

Thus, an extension is obtained by adding one of the largest sets of hypotheses to F while remaining consistent. Note that if F is consistent, then there is always an extension, which is not the case in DL. Intuitively, E is stable if each time 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 (and to stable models of ASP). They represent stable configurations of BDSs. Ghost extensions do not have any correspondence in DL. While default theories may have no extensions, this is not the case in hypothesis theories. Nevertheless, default theories that have no extensions seem to contain what looks very much like stable or unstable cycles. In the sequel, by using $\mathcal{H}$, we show that stable cycles are characterized by ghost extensions. Moreover, specific unstable cycles can also be characterized by ghost extensions, but the generalization of this result is a conjecture yet.

4 Representing Interaction Graphs into $\mathcal{H}$

We saw previously that a genetic network modeled by a BDS can be represented by both an ATG and an IG. As told, an important and open question concerns the formal links between these two representations. This section deals only with IGs. Let G be an arbitrary IG. Such a graph is translated into a hypothesis theory $\mathcal{T}(G)$ and properties related to extensions of $\mathcal{T}(G)$ are then proved. These properties will be used in the next section, devoted to ATG: G will be the IG related to an ATG, while both stable configurations and stable cycles of the ATG are studied thanks to $\mathcal{T}(G)$. Theorem 3 states that there exists an isomorphism between stable configurations of $\mathscr{G}$ and stable extensions of $\mathcal{T}(G)$ and Theorem 4 states that every negative feedback circuit corresponds to a set of equivalent ghost extensions.

[2] For a classical lecture on modal logics, see for instance (Chellas 1980), among others.

4.1 Representation

One of the interests of hypothesis logic is that this bimodal logic enables us to use three kinds of information: i, $\mathrm{L}i$ and $\mathrm{H}i$. This increasing of expressive power allows a more precise representation of biological networks. Hence, by combining modalities with negations, we can use $\{i, \mathrm{H}i, \mathrm{H}\neg i, \mathrm{L}i, \mathrm{L}\neg i\}$.

Given an IG G of a BDS modeling the genetic network of a cell, and i a protein, using hypothesis logic, we define that:

- i means that *i is present in the cell and $\neg i$ means that it is absent.*
- $\mathrm{L}i$ means that *i is produced by the cell (i is being activated) and $\neg\mathrm{L}i$ means that i is not produced (i is not being activated).*
- $\mathrm{L}\neg i$ means that *i is destroyed by the cell (i is being inhibited) and $\neg\mathrm{L}\neg i$ means that i is not destroyed (i is not being inhibited).*
- $\mathrm{H}i$ (resp. $\neg\mathrm{H}i$) 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.*
- $\mathrm{H}\neg i$ (resp. $\neg\mathrm{H}\neg i$) 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 $\mathrm{H}i$ 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 $\mathrm{H}\neg i$ gives the authorization to destroy i.

Meanwhile, it is important to note that $\mathrm{L}i$ and $\mathrm{L}\neg i$ are actually actions (production or destruction of a protein). So there is a difference between $\mathrm{L}\neg i$ which says that i is destroyed, and $\neg\mathrm{L}i$ which says that i is not produced, and hence is weaker. Likewise, there is a similar distinction between $\mathrm{H}\neg i$ and $\neg\mathrm{H}i$.

We first focus on some important properties of our translation with respect to genetic networks[3].

Proposition 1 *If G is the IG of a BDS modeling a genetic network and if i is a protein, the following holds in $\mathcal{H}$:*

(1) $\mathrm{L}i \rightarrow i$ *and* $\mathrm{L}\neg i \rightarrow \neg i$ *(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* $\neg(\mathrm{L}\neg i \wedge \mathrm{H}i)$ *(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) $\neg(\mathrm{L}i \wedge L\neg i)$ *(i.e. it is impossible to produce and destroy i at the same time.)*

4.2 Translation of an interaction graph.

An IG $G = (V, I)$ is translated into a hypothesis theory $\mathcal{T}(G) = \{\mathrm{HY}(G), \mathrm{F}(G)\}$ so that every arc $(i, s, j) \in I$ is translated into a pair of implications of $\mathcal{H}$. More precisely:

- A positive arc $(i, +, j)$ is translated into: $\{\mathrm{H}i \rightarrow \mathrm{L}j, \mathrm{H}\neg i \rightarrow \mathrm{L}\neg j\}$.
- A negative arc $(i, -, j)$ is translated into: $\{\mathrm{H}i \rightarrow \mathrm{L}\neg j, \mathrm{H}\neg i \rightarrow \mathrm{L}j\}$.
- $\mathrm{F}(G)$ is the union of the translations of all elements of I.
- $\mathrm{HY}(G)$ is the set of all $\mathrm{H}i$ and $\mathrm{H}\neg i$ appearing in $\mathrm{F}(G)$.

Note 3 *This translation only uses implications between two atomic formulas. These implications could be considered as binary clauses. Therefore only a fragment of the plain formalism $\mathcal{H}$ is used in this paper, which is enough for the description of "conventional" BDSs. Note that $\mathcal{H}$ formulas may contain all the logical connectors $(\wedge, \vee, \neg, \rightarrow, \leftrightarrow \ ...)$ hence full $\mathcal{H}$ can be used to describe other properties of biological networks, e.g. the binding (two proteins bind to give a new protein). It is also possible to assert the proposition i alone or* $\mathrm{L}i$ *alone or* $\mathrm{H}i$ *alone. We can even avoid the double implication $\{Hx \rightarrow Ly, H\neg x \rightarrow L\neg y\}$ given for the translation of a BDS: for some functions only one involvement of the two can be enough. This increases the expressive power of the formalism, which in turn should increase the algorithmic complexity, but $\mathcal{H}$ is still usable however.*

The following definitions and propositions, are needed for understanding the intuition behind this representation of IGs by $\mathcal{H}$. They will especially allow us to state properties and theorems 3 and 4 which make links between IGs and ATGs. In $\mathcal{H}$, it is usually allowed to have both $\mathrm{H}i$ and $\mathrm{H}\neg i$. Regarding the fragment of $\mathcal{H}$ used here, Proposition 2 below shows that this no longer holds, because of the double logical implication obtained from our translation of an arc:

Proposition 2 *Let $G = (V, I)$ an IG and $i \in V$. For every* $\mathrm{H}i$ *of $\mathcal{T}(G)$, $\neg(\mathrm{H}i \wedge \mathrm{H}\neg i)$ holds.*

Definition 2 *Let $G = (V, I)$ an IG such that $V = \{1, .., n\}$. Let $\mathcal{T}(G) = \{\mathrm{HY}(G), \mathrm{F}(G)\}$ the translation of G into a hypothesis theory. Let $E = \{\mathrm{Th}(\mathrm{F}(G)) \cup \{\mathrm{H}y_k\}\}$ be an extension of $\mathcal{T}(G)$ obtained by adding to $\mathrm{F}(G)$ a maximal consistent set $\{\mathrm{H}y_k\}$ of hypotheses, with $y_k \in \{1, ..., n, \neg 1, ..., \neg n\}$. (For lightening the reading, we simply write that E is an extension of G). We have:*

1. E is complete *if, for all $i \in V$,* $\mathrm{H}i \in E$ *or* $\mathrm{H}\neg i \in E$.

2. A vertex $i \in V$ is free *in E if* $\mathrm{L}i \notin E$ *and* $\mathrm{L}\neg i \notin E$. *It is* fixed *otherwise.*

3. The degree of freedom of E, denoted $\deg(E)$*, is the number of free vertices that compose it.*

4. The mirror *of E, denoted* $\mathrm{mir}(E)$*, is defined as* $\mathrm{mir}(E) = \mathrm{Th}(\mathrm{F}(G) \cup \{\mathrm{H}\neg y_k\})$.

5. The generating set *of E, denoted* $\mathrm{Gen}(E)$ *is the set of formulas* $(\mathrm{H}y \rightarrow \mathrm{L}z) \in \mathrm{F}(G)$ *such that* $\mathrm{H}y \in \{Hy_k\}$.

6. The graph of E*, denoted by $G(E)$, is the unsigned digraph of vertices $\{y_1, ..., y_m\}$ such that:*

$$(y_i, y_j) \in G(E) \iff (\mathrm{H}y_i \rightarrow \mathrm{L}y_j) \in \mathrm{Gen}(E).$$

Proposition 3 *Let G be an IG and E one of its extensions. The mirror of E is also an extension of G.*

Intuition might suggest that the notions of stable extension, complete extension, and extension of degree 0 are equivalent. In fact this is wrong in the general case and it is only possible to prove that a stable extension is complete.

[3] Note to the rewievers: all the proofs of the original propositions and theorems given in the paper can be found at the url `https://amubox.univ-amu.fr/index.php/s/nhwsZ5eqV8BYVv1`

But we prove that, when any vertex of the IG G has an incoming arc, if E is complete then E is both stable and of degree 0; this is especially the case for circuits.

Proposition 4 *Let $G = (V, I)$ be an IG of a given BDS f and let E be an extension of G. The following holds:*

1. If E is stable, then E is complete.

Moreover, if each of the vertices of G has at least one incoming arc:

2. If E is complete, then $\deg(E) = 0$.

3. If E is complete, then E is stable.

4. If E is stable then $\deg(E) = 0$.

Proposition 5 *Let $E = \{\mathrm{Th}(\mathrm{F}(G)) \cup \{\mathrm{H}y_k\}\}$ be an extension and consider $\mathrm{Gen}(E)$ its generating set:*

1. $E = \{\mathrm{Th}(\mathrm{Gen}(E)) \cup \{\mathrm{H}y_k\}\}$.

2. If $\mathrm{H}x \in \{\mathrm{H}y_k\}$ then $\mathrm{Gen}(E)$ cannot contain both $\mathrm{H}x \to \mathrm{L}y$ and $\mathrm{H}\neg x \to \mathrm{L}\neg y$ at the same time.

3. If $\deg(E) = 1$ then an order can be chosen among the y_k with a circular permutation such that: $\mathrm{Gen}(E) = \{\mathrm{H}i \to \mathrm{L}(i+1), \mathrm{H}(i+1) \to \mathrm{L}(i+2), ...\mathrm{H}(i-2) \to \mathrm{L}(i-1)\}$.

Proposition 6 *Given $G = (V, I)$, with $V = \{1, ..., n\}$, a negative circuit of size n, then:*

1. $\mathcal{T}(G)$ has no extensions of degree 0.

2. $\mathcal{T}(G)$ has $2n$ extensions of degree 1. We will say that these 2n extensions are equivalent.

The examples below serve as an illustration of the notions introduced here.

Example 2 (continued) Consider the BDS of the function $f(x_1, x_2, x_3, x_4) = (\neg x_4, x_1, \neg x_2, x_3)$ studied in Example 2 and Figure 3. This BDS corresponds to a positive circuit. Its admits two stable configurations, $(x_1, x_2, \neg x_3, \neg x_4)$ and $(\neg x_1, \neg x_2, x_3, x_4)$ and an unstable oscillation. Let $G(f)$ the IG of f, depicted in Figure 3-b. By construction $G(f)$ is a set of four arcs:

- $G(f) = \{(1, +, 2), (2, -, 3), (3, +, 4), (4, -, 1)\}$

The positive arc $(1, +, 2)$ is translated into $\mathcal{H}$ by the pair of formulas $\{\mathrm{H}1 \to \mathrm{L}2, \mathrm{H}\neg 1 \to \mathrm{L}\neg 2\}$ and the negative arc $(2, -, 3)$ is translated by the pair $\{\mathrm{H}2 \to \mathrm{L}\neg 3, \mathrm{H}\neg 2 \to \mathrm{L}3\}$. The other arcs are translated in the same way, therefore the translation of $G(f)$ into a hypothesis theory is $\mathcal{T}(G(f)) = \{\mathrm{HY}(G(f)), \mathrm{F}(G(f))\}$, where:

- $\mathrm{HY}(G(f)) = \{\mathrm{H}1, \mathrm{H}2, \mathrm{H}3, \mathrm{H}4, \mathrm{H}\neg 1, \mathrm{H}\neg 2, \mathrm{H}\neg 1, \mathrm{H}\neg 4\}$
- $\mathrm{F}(G(f)) = \{\mathrm{H}1 \to \mathrm{L}2, \mathrm{H}\neg 1 \to \mathrm{L}\neg 2, \mathrm{H}2 \to \mathrm{L}\neg 3, \mathrm{H}\neg 2 \to \mathrm{L}3, \mathrm{H}3 \to \mathrm{L}4, \mathrm{H}\neg 3 \to \mathrm{L}\neg 4, \mathrm{H}4 \to \mathrm{L}\neg 1, \mathrm{H}\neg 4 \to \mathrm{L}1\}$

We can show that $\mathcal{T}(G(f))$ has two stable extensions: $E1$ obtained by adding to $F(G(f))$ the set of hypothesis $\{\mathrm{H}1, \mathrm{H}2, \mathrm{H}\neg 3, \mathrm{H}\neg 4\}$ and $E2$ obtained by adding to $F(G(f))$ the set of hypotheses $\{\mathrm{H}\neg 1, \mathrm{H}\neg 2, \mathrm{H}3, \mathrm{H}4\}$[4]. These extensions correspond to the two stables configurations of the related BDS.

- $E1 = \mathrm{Th}(\mathrm{F}(G(f)) \cup \{\mathrm{H}1, \mathrm{H}2, \mathrm{H}\neg 3, \mathrm{H}\neg 4\})$,
- $E2 = \mathrm{Th}(\mathrm{F}(G(f)) \cup \{\mathrm{H}\neg 1, \mathrm{H}\neg 2, \mathrm{H}3, \mathrm{H}4\})$.

In the sequel, given i a proposition of propositional calculus, we will consider that i, Hi,Li, $\neg i$, $\neg Hi$ et $\neg Li$ are literals of $\mathcal{H}$, and that $\mathrm{H}i \to \mathrm{L}j$ is a clause. From the deductive closure of $E1$ with subsumption, we obtain that these extensions are logically equivalent to sets of literals:

- $E1 = \mathrm{Th}\{\mathrm{H}1, \mathrm{H}2, \mathrm{H}\neg 3, \mathrm{H}\neg 4, \neg\mathrm{H}\neg 1, \neg\mathrm{H}\neg 2, \neg\mathrm{H}3, \neg\mathrm{H}4, \mathrm{L}1, \mathrm{L}2, \mathrm{L}\neg 3, \mathrm{L}\neg 4, \neg\mathrm{L}\neg 1, \neg\mathrm{L}\neg 2, \neg\mathrm{L}3, \neg\mathrm{L}4, 1, 2, \neg 3, \neg 4\}$
- $E2 = \mathrm{Th}\{\mathrm{H}\neg 1, \mathrm{H}\neg 2, \mathrm{H}3, \mathrm{H}4, \neg\mathrm{H}1, \neg\mathrm{H}2, \neg\mathrm{H}\neg 3, \neg\mathrm{H}\neg 4, \mathrm{L}\neg 1, \mathrm{L}\neg 2, \mathrm{L}3, \mathrm{L}4, \neg\mathrm{L}1, \neg\mathrm{L}2, \neg\mathrm{L}\neg 3, \neg\mathrm{L}\neg 4, \neg 1, \neg 2, 3, 4\}$

For the sake of simplicity, let us assimilate any extension $\mathrm{Th}(\mathrm{F}(G(f)) \cup \{\mathrm{H}1, ..., \mathrm{H}n\})$ with the set of hypotheses $\{\mathrm{H}1, ..., \mathrm{H}n\}$ associated with it. Looking at $E1$ we notice that, in accordance with Definitions 1, and 2, we have:

- $E1$ and $E2$ are stable extensions because for all i, $\neg\mathrm{H}i \in E1$ (resp $E2$) $\Rightarrow L\neg i \in E1$ (resp $E2$).
- $E1$ is complete because for all i, either Hi belongs to $E1$ or $H\neg i$ belongs to $E1$.
- For all i, $Li \in E1$ or $L\neg i \in E1$. So all vertices are fixed, and the degree of freedom of $E1$ is 0.
- $E2$ is the mirror of $E1$.
- The generating set of $E1$, is $Gen(E1) = \mathrm{H}1 \to \mathrm{L}2, \mathrm{H}2 \to \mathrm{L}\neg 3, \mathrm{H}\neg 3 \to \mathrm{L}\neg 4, \mathrm{H}\neg 4 \to \mathrm{L}1\}$.
- The graph of $E1$, depicted in Figure 5-c, is: $G(E) = (1, 2), (2, \neg 3), (\neg 3, \neg 4), (\neg 4, 1)$.

From the biological side, the subset $\{\mathrm{L}1, \mathrm{L}2, \mathrm{L}\neg 3, \mathrm{L}\neg 4\}$ of $E1$ represents the expression pattern of each protein in $E1$: 1 and 2 are produced by the cell and 3 and 4 are destroyed. Moreover the subset $\{1, 2, \neg 3, \neg 4\}$ of $E1$ (in fact the vertices of $G(E)$) gives the status of each proteins: 1 and 2 are present in the cell and $\neg 3$ and $\neg 4$ are absent. Similarly in $E2$, the subsets $\{\neg\mathrm{L}1, \neg\mathrm{L}2, \mathrm{L}3, \mathrm{L}4\}$ and $\{\neg 1, \neg 2, 3, 4\}$ represent the expression pattern and the status of proteins.

An intuition of the computation of $E1$ is given by the construction process described by Figure 5. Figure 5-a is the IG of f. Figure 5-b gives the construction of $E1$. At first $E1$ is empty. We add to $E1$ the hypothesis $\mathrm{H}1 \in HY(G(f))$. Since $(\mathrm{H}1 \to \mathrm{L}2) \in F(G(f))$ we get L2. The instance $(\mathrm{L}2 \to \neg H\neg 2)$ of the axiom LI of $\mathcal{H}$ then tells that it is impossible to have $H\neg 2$. Construction of $E1$ goes on by adding the hypothesis H2 to $E1$. We get $\mathrm{L}\neg 3$ with $\mathrm{H}2 \to \mathrm{L}\neg 3 \in F(G(f)$. The axiom LI gives $\neg\mathrm{H}3$ and we add the hypotheses $\mathrm{H}\neg 3$ which gives $\mathrm{L}\neg 4$. We end up by adding $\mathrm{H}\neg 4$ which gives L1.

By only looking at i we have $G(E1)$, the graph of $E1$ (Figure 5-c) which represents the first stable configuration of f. In this final graph, each arc between 2 vertices denotes the relation of causality that links the corresponding proteins: for 1 to be present we need 4 to be absent, that is 3 to be absent in its turn, which is involved by the presence of 2, itself caused by the presence of 1 and so on... In a similar way, we build the extension $E2$ starting from $H\neg 1$.

Example 3 (continued) Consider the BDS of the function $g(x_1, x_2, x_3, x_4) = (\neg x_4, x_1, x_2, x_3)$ studied in Example 3. It admits one attractor (a stable cycle of length 8). It admits also an unstable cycle of length 8. Let be $G(g)$ the IG of g. By construction it is a set of four arcs:

- $G(g) = \{(1, +, 2), (2, +, 3), (3, +, 4), (4, -, 1)\}$

Following the same guidelines as above, the translation of $G(g)$ into a hypothesis theory is $\mathcal{T}(G(g))$ gives us here eight ghost extensions. The first one is:

[4] This is shown by attempting to add to $\mathrm{F}(G(f))$ each subset of $\mathrm{HY}(G(f))$ and keeping only those which are the maximals ones consistent with $\mathrm{F}(G(f))$.

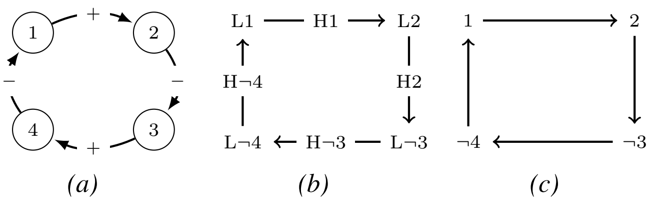


Figure 5: *(a)* IG $G(f)$, *(b)* Construction of extension $E1$ and *(c)* $G(E1)$ (the graph of $E1$) of the positive circuit defined in Example 2.

• $E1 = \mathrm{Th}(\mathrm{F}(G(g)) \cup \{\mathrm{H1, H2, H3}\})$
$= \mathrm{Th}(\{\mathrm{H1, H2, H3, \neg H\neg 1, \neg H\neg 2, \neg H\neg 3, \neg H\neg 4,}$
$\mathrm{\neg H4, L2, L3, L4, \neg L\neg 1, \neg L\neg 2, \neg L3, \neg L\neg 4}, 2, 3, 4\})$.

The generator set of $E1$ is $\{\mathrm{H1{\to}L2, H2{\to}L3, H3{\to}L4}\}$. The literal $\neg\mathrm{H}\neg 1 \in E1$ characterizes $E1$ as a ghost extension because $L1 \notin E1$. The extension is not complete because it contains neither H4 nor H¬4. Vertex 1 is free because $E1$ contains neither L1 nor L¬1. This is the only free vertex, hence $E1$ is a ghost extension of degree 1. For this extension, vertices 2, 3 and 4 are fixed while nothing is known about 1. The drawing of $E1$ (Figure 6-b) shows what happens. Similary to Example 2, $E1$ is constructed by adding {H1, H2, H3} to $\mathrm{F}(G(g))$. The set $\{\mathrm{L2, L3, L4}\}$ is obtained, which yields $\{2, 3, 4\}$ from axiom (T). Hence it is impossible to add H¬4 because $(\mathrm{H\neg 4 \to \neg L4})$ from the axiom of coherence and $\mathrm{L4} \in E1$. Also, one can not add H4 because $(\mathrm{H4 \to L\neg 1}) \in \mathrm{F}(G(g))$ and $\mathrm{L1} \in E1$, which will imply 1 and ¬1 with axiom (T). In Figure 6-b, the place for 1 is empty because the extension cannot contain both L1 and L¬1, and 1 is free. Indeed, in order to get L1, we should use $\mathrm{H\neg 4 \to L1}$, which is impossible because L4 is true and, because from the axiom of coherence, $\mathrm{L4 \to \neg H\neg 4}$. We cannot have L¬1 because H1 is true and $\mathrm{H1 \to \neg L\neg 1}$. The graph of $E1$, $G(E1)$ is depicted in Figure 6-c.

It is important to see that the notion of degree of freedom plays a key role here. Noting that Figure 6-b (resp. Figure 6-c) are not circuits because there is no arc between $L4$ (resp. 4) and the place of 1. From $L4$ we can then *escape* the incomplete circuit. Let us indeed show how to do it. Since $E1$ is an extension, $E2$ the mirror of $E1$ is also an extension from Proposition 3 (see Figure 7). Taking into account that 1 is free in $E1$ and also in $E2$, $E1$ and in $E2$ can be connected ($E1 \rightleftharpoons E2$) by binding L4 to L¬1 using H¬4 on one hand, and L¬4 to L1 using H4 on the other hand (Figure 8-a). By linking $G(E1)$ and $G(E2)$ (Figure 8-b), we then obtain a cycle $(1, 2, 3, 4, \neg 1, \neg 2, \neg 3, \neg 4)$ of length 8 such as depicted. This cycle represents the expression pattern of each protein over time: we turn two times in the IG until returning to the initial state.

In this example, there are actually eight equivalent extensions $E1 = \{\mathrm{L2, L3, L4}\}$, its miror $E2 = \{\mathrm{L\neg 2, L\neg 3, L\neg 4}\}$ and 6 other extensions which come from permutations on i, namely:
$\{\mathrm{L3, L4, L\neg 1}\}, \{\mathrm{L4, L\neg 1, L\neg 2}\}, \{\mathrm{L\neg 1, L\neg 2, L\neg 3}\},$
$\{\mathrm{L\neg 2, L\neg 3, L\neg 4}\}, \{\mathrm{L\neg 3, L\neg 4, L1}\}, \{\mathrm{L\neg 4, L1, L2}\}.$

Note that each of these extensions corresponds to 3 successive entities of the cycle above. The latter cycles contains implicitly each of the 8 possible extensions. There is a trick here since we actually used an update that is not asynchronous, but this trick, aiming at authorizing synchronous changes appears to be virtuous. It will be seen that these 8 extensions correspond to the stable cycle of the ATG $\mathscr{G}(g)$.

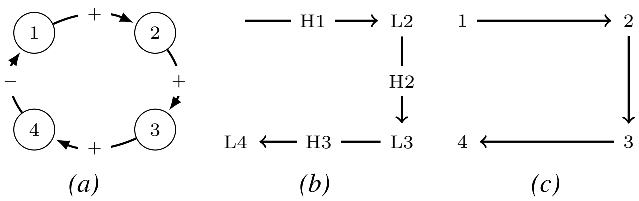


Figure 6: *(a)* Interaction graph $G(g)$, *(b)* Construction of extension $E1$ and *(c)* $G(E1)$ (graph of $E1$) of the negative circuit defined in Example 3 (continued).

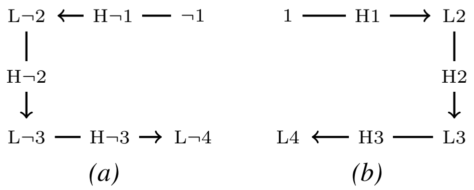


Figure 7: *(a)* $E1$ and *(b)* his mirror $E2$ of the negative circuit defined in Example 3 (continued)

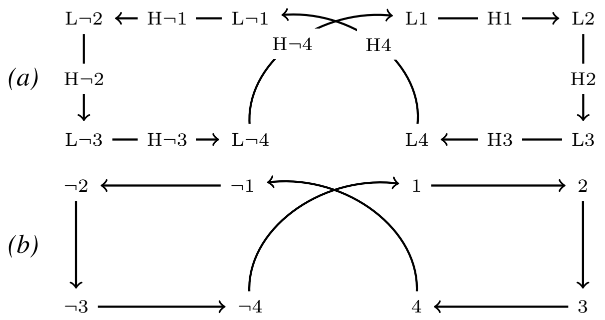


Figure 8: *(a)* Construction of $E1 \rightleftharpoons E2$ and *(b)* its graph $G(E1 \rightleftharpoons E2)$ of the negative circuit defined in Example 3

5 Representing Asynchronous Transition Graphs into $\mathcal{H}$

Consider an asynchronous BDS, whose its IG is $G = (V, I)$ and its ATG is $\mathscr{G}$. Let $\mathcal{T}(G) = \{\mathrm{HY}(G), \mathrm{F}(G)\}$ be the hypothesis theory associated with G. Up to now, we have studied the representation of IGs into hypothesis logic.

This section studies the relationship between this representation and the ATG. It uses *Kripke semantics* (Kripke 1963) that has been defined for normal modal logics, *i.e.*, the logics that contain at least axiom (K). We only provide here the bases needed for our developments. A *Kripke structure* is a digraph $K = (W, R)$ where W (the universe) is a set $\{w_k\}$ of worlds and $R \subseteq W \times W$ is a binary relation among worlds: the *accessibility relation*. When $w_k\, R\, w'_k$, w'_k is *accessible* from w_k. A *Kripke model* is obtained by assigning in every world a truth value to every proposition i. A world is then mapped to a logical interpretation, and hence implicitely to a state of a BDS. Modal formulas other than i 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 reachable worlds from w. The different axioms that hold in different modal logics depend on the properties of the accessibility relations R. It is known that:

1. For the modal system K, R is any relation.
2. Axiom (T) holds if and only if R is reflexive.

Example 4 *Consider the Kripke structure K such that $K = (W, R)$, where $W = \{w_1, w_2, w_3\}$ and $R = \{(w_1, w_2), (w_1, w_1), (w_2, w_2), (w_3, w_3)\}$.*

Consider now that a truth value is assigned to three variables in each world of the universe W such that: $w_1 = \{1, 2, 3\}$, $w_2 = \{\neg 1, 2, 3\}$, $w_3 = \{\neg 1, \neg 2, 3\}$. Let us add modal formulas with respect to the definition of Krypke semantic. For example in w_1, L2 is true because 2 is in both w_1 and w_2 (the worlds that are reachable from w_1). For the sake of clarity, we do not put in this schema the negations of modal formulas. Now, consider $\mathcal{H}$. If w_1 was an extension, then it would be a ghost extension of degree 1, because L2 and L3 are true while neither L1 nor L¬1 are. Similarly, for both worlds w_2 and w_3, if they were extensions, they would be stable extensions of degree 0 because for all $i \in \{1, 2, 3\}$, either Li or L¬i is true; they would then correspond to stable configurations of a related BDS because they do not admit any outward arcs.

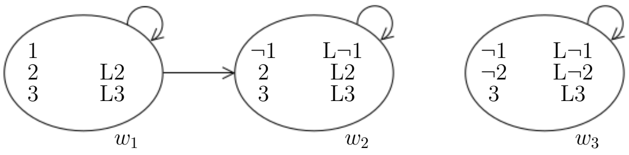


Figure 9: Kripke universe discussed in Example 4.

We use the Kripke structure $K = (W, R)$ whose finite set of worlds W is the set of all interpretations (called *canonical universe*) such that $w_k\, R\, w'_k$ if and only if w'_k is reachable from w_k and differs from w_k by one and only one proposition. Under these conditions, the ATG of any BDS is a Kripke structure. Since the modal system T is reflexive, loops appear on every world. Given such a framework, for any world w_k and any entity x, Lx = true if and only if x = true in every w'_k reachable from w_k.

Example 1-continued Consider the previous function $g(x_1, x_2, x_3) = (\neg x_3, x_1, x_2)$ of Example 1. Similarly to Example 3-continued, constructing the related hypothesis theory leads the following set of formulas:

$$\mathrm{F}(G(g)) = \{\mathrm{H}1 \rightarrow \mathrm{L}2, \mathrm{H}2 \rightarrow \mathrm{L}3, \mathrm{H}3 \rightarrow \mathrm{L}\neg 1, \mathrm{H}\neg 1 \rightarrow \mathrm{L}\neg 2, \mathrm{H}\neg 2 \rightarrow \mathrm{L}\neg 3, \mathrm{H}\neg 3 \rightarrow \mathrm{L}1\},$$

which allows to obtain the following 6 equivalent extensions, by focusing only on the Li and L¬i which are true:

$E1 = (\mathrm{L}2, \mathrm{L}3), E2 = (\mathrm{L}\neg 1, \mathrm{L}3), E3 = (\mathrm{L}\neg 1, \mathrm{L}\neg 2),$
$E4 = (\mathrm{L}\neg 2, \mathrm{L}\neg 3), E5 = (\mathrm{L}1, \mathrm{L}\neg 3), E6 = (\mathrm{L}1, \mathrm{L}2).$

Figure 9 is a simplified representation of the Krypke model associated with the theory $F(G(g))$. The 8 vertices are the worlds, and the arrows express the accessibility relation. There is a loop on each vertex because (T) is reflexive. The 6 extensions $E1, .., E6$ are represented by the 6 vertices whose degree of freedom is minimal (that is of degree 1). The other two vertices $\{1, \neg 2, 3\}$ and $\{\neg 1, 2, \neg 3\}$ are not extensions because they are not maximal. Note that these 6 extensions correspond to the 6 configurations of the stable cycle of the corresponding ATG of function g given in example 1 (Figure 11). Note also that the set of arrows that represent the accessibility relation contains the set of arrows that represent the ATG transitions of g. The missing arrows are the loops that represent the reflexivity. By taking in account that the degree of a world is a natural generalization of degree of an extension, we get the following results.

Proposition 7 *Let $K = (W, R)$ be a Kripke structure associated with an IG and its underlying hypothesis theory. If $w_k \in W$ is a world of degree δ, there are exactly δ worlds, different from w_k, reachable from w_k*

Theorem 3 *Let $\mathscr{S}$ be an asynchronous BDS, G and $\mathscr{G}$ the corresponding IG and ATGs and $\mathcal{T}(G)$ the hypothesis theory related to G.*

1. If E is a stable extension of $\mathcal{T}(G)$ and if $\{\mathrm{L}y_1, ..., \mathrm{L}y_n\}$ is the set of all Li and L¬i that are true in E, and if the x_i are the Boolean values of the y_i, then $\{x_1, ..., x_n\}$ is a stable configuration of $\mathscr{S}$.

2. $\{x_1, ..., x_n\}$ is a stable configuration of $\mathcal{T}(G)$, then there exists a stable extension E of $\mathscr{G}$ that contains $\{\mathrm{L}y_1, ..., \mathrm{L}y_n\}$, such that $y_i = i$ (resp $y_i = \neg i$) il $x_i = 1$, (resp. $x_i = 0$).

Theorem 4 *Let $\mathscr{S}$ be an asynchronous BDS of ATG $\mathscr{G}$, whose IG G is a negative circuit. Let E1 be a ghost extension of $\mathcal{T}(G)$. The set of all extensions equivalent to E1 corresponds to a stable cycle of $\mathscr{G}$.*

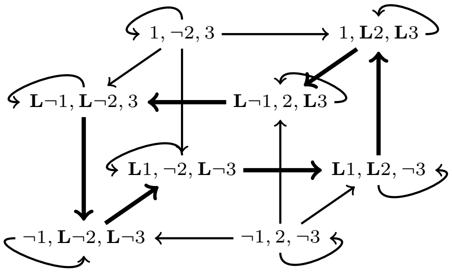


Figure 10: Krypke representation of F$(G(g))$.

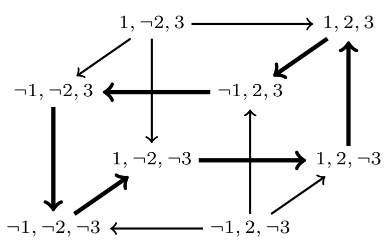


Figure 11: ATG of function g given Example 1.

Note 4 With same arguments as those used for the proof of proposition 7, if $deg(E){>}1$ the Kripke model gives at least one possibility to exit from a cycle. The cycle is then unstable.

6 Conclusion

This paper is an attempt for representing BDSs into Hypothesis Logics; the difficulty is to find how to represent the dynamics. There is still much to study, especially regarding a generalization of Theorem 4 to unstable cycles. Note 4 above is a hint for such a study. Another perspective is the validation of BDS representation in $\mathcal{H}$ by the obtaining of fundamental theorems. The fact that the logical representation of a positive circuit has two stable mirror extensions and, that a negative circuit is equivalent to a single set of $2n$ equivalents ghost extensions is a step towards this validation because it corresponds to the esults set in (Remy et al. 2003). The case of synchronous transitions remains under consideration for a further study. 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 and constructive (which is not the case for DL and ASP).

References

Akman, O. E.; Watterson, S.; Parton, A.; Binns, N.; Millar, A. J.; and Ghazal, P. 2012. Digital clocks: simple Boolean models can quantitatively describe circadian systems. *Journal of The Royal Society Interface*.

Aracena, J.; González, M.; Zuñiga, A.; Mendez, M. A.; and Cambiazo, V. 2006. Regulatory network for cell shape changes during Drosophila ventral furrow formation. *Journal of Theoretical Biology* 239:49–62.

Chellas, B. 1980. *Modal logic, an introduction*. Cambridge University Press.

Davidich, M. I., and Bornholdt, S. 2008. Boolean network model predicts cell cycle sequence of fission yeast. *PLoS One* 3:e1672.

Demongeot, J.; Goles, E.; Morvan, M.; Noual, M.; and Sené, S. 2010. Attraction basins as gauges of the robustness against boundary conditions in biological complex systems. *PLoS One* 5:e11793 (18 pages).

Demongeot, J.; Elena, A.; Noual, M.; Sené, S.; and Thuderoz, F. 2011. "immunetworks", intersecting circuits and dynamics. *Journal of Theoretical Biology* 280:19–33.

Demongeot, J.; Noual, M.; and Sené, S. 2012. Combinatorics of Boolean automata circuits dynamics. *Discrete Applied Mathematics* 160:398–415.

Doncescu, A., and Siegel, P. 2015. *Emerging Trends in Computational Biology, Bioinformatics, and Systems Biology*. Elsevier. chapter DNA double-strand break-based nonmonotonic logic, 409–427.

Doncescu, A.; Siegel, P.; and Le, T. 2014. Representation and efficient algorithms for the study of cell signaling pathways. In *Proceedings of ICAI'2014*, 504–510. IEEE Computer Society.

Fauré, A.; Naldi, A.; Chaouyia, C.; and Thieffry, D. 2006. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics* 22:e124–e131.

Jacob, F., and Monod, J. 1978. Genetic regulation mechanisms in the synthesis of proteins. In *Selected Papers in Molecular Biology by J. Monod*, 433–471. Academic Press.

Kauffman, S. A.; Peterson, C.; Samuelsson, B.; and Troein, C. 2003. Random Boolean network models and the yeast transcriptional network. *PNAS* 100:14796–14799.

Kauffman, S. A. 1969. Metabolic stability and epigenesis in randomy constructed nets. *Journal of Theoretical Biology* 22:437–467.

Kripke, S. A. 1963. Semantical analysis of modal logic I Normal modal propositional calculi. *Mathematical Logic Quarterly* 9:67–96.

Li, F.; Long, T.; Lu, Y.; Ouyang, Q.; and Tang, C. 2004. . *PNAS* 101:4781–4786.

Lifschitz, V. 1999. *The Logic Programming Paradigm: A 25-Year Perspective*. Springer. chapter Action languages, answer sets, and planning, 357–373.

Lukaszewicz, W. 1988. Considerations on Default Logic – An Alternative Approach. *Computational Intelligence* 4:1–16.

Melliti, T.; Regnault, D.; Richard, A.; and Sené, S. 2013. On the convergence of Boolean automata networks without negative cycles. In *Proceedings of AUTOMATA'2013*, volume 8155 of *LNCS*, 124–138. Springer.

Melliti, T.; Noual, M.; Regnault, D.; Sené, S.; and Sobieraj, J. 2015. Asynchronous dynamics of Boolean automata double-cycles. In *Proceedings of UCNC'2015*, volume 9252 of *LNCS*, 250–262. Springer.

Mendoza, L.; Thieffry, D.; and Alvarez-Buylla, E. R. 1999. Genetic control of flower morphogenesis in *Arabidopsis thaliana*. *Bioinformatics* 15:593–606.

Reiter, R. 1980. A logic for default reasoning. *Artificial Intelligence* 13:81–132.

Remy, E.; Mossé, B.; Chaouyia, C.; and Thieffry, D. 2003. A description of dynamical graphs associated to elementary regulatory circuits. *Bioinformatics* 19:ii172–ii178.

Remy, E.; Ruet, P.; and Thieffry, D. 2008. Graphic requirement for multistability and attractive cycles in a Boolean dynamical framework. *Advances in Applied Mathematics* 41:335–350.

Richard, A., and Comet, J.-P. 2007. Necessary conditions for multistationarity in discrete dynamical systems. *Discrete Applied Mathematics* 155:2403–2413.

Richard, A. 2010. Negative circuits and sustained oscillations in asynchronous automata networks. *Advances in Applied Mathematics* 44:378–392.

Robert, F. 1986. *Discrete Iterations: A Metric Study*. Springer.

Roenneberg, T., and Merrow, M. 2003. The network of time: understanding the molecular circadian system. *Current Biology* 13:R198–R207.

Schwind, C., and Siegel, P. 1994. A modal logic for hypothesis theory. *Fundamenta Informaticae* 21:89–101.

Siegel, P., and Schwind, C. 1993. Modal logic based theory for non-monotonic reasonning. *Journal of Applied Non-classical Logic* 3:73–92.

Siegel, P.; Doncescu, A.; Risch, V.; and Sené, S. 2017. Vers une représentation des systèmes dynamiques discrets en logique des hypothèses *. Journées d'Intelligence Artificielle Fondamentale, Jul. 2017, Caen, France, hal-01566164 .

Thomas, R. 1973. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology* 42:563–585.

Thomas, R. 1981. *Numerical Methods in the Study of Critical Phenomena*. Springer. chapter On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations, 180–193.