Representation of gene regulation networks by hypothesis logic-based Boolean systems

Pierre Siegel, Andrei Doncescu, Vincent Risch, Sylvain Sené

Representation of Gene Regulation Networks by Hypothesis Logic Based Boolean Systems

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

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

Abstract. Boolean Dynamical Systems (BDS) are systems of entities described by Boolean variables providing interaction in discrete time. They are particularly used in the modeling of gene signaling pathways. We present new representations of BDSs and of gene regulation networks, using a modal non-monotonic logic ($\mathcal{H}$) By using these representations every Boolean network can be represented by a set of modal formulas of $\mathcal{H}$, and therefore by a set of Kripke models of $\mathcal{H}$. The study of a BDS focuses in particular on the search of its stable configurations, limit cycles and unstable cycles. By using our representation, we prove that it is possible to discriminate between stable configurations, limit cycles and unstable cycles thanks to the introduction of a new concept, namely the *ghost extensions*. In addition the formalism introduced in this article uses a minimalist definition of the language of $\mathcal{H}$, but sufficient to represent BNs. This restriction allows us to translate $\mathcal{H}$ into propositional calculus, hence to use SAT algorithms, and therefore to benefit of a simple and powerful implementation.

Keywords: Gene regulation, Genetic networks, Boolean networks, Non-monotonic logic, Modal logic, Hypothesis logic, Computational Systems Biology, SAT algorithms.

1 Introduction

This article gives formal links between Boolean Dynamical Systems (BDSs) and a modal non-monotonic logic called *hypothesis logic* ($\mathcal{H}$) [33, 36] with an application to Boolean gene regulation networks, thanks to the introduction of a new concept, namely the *ghost extensions*. We also also show that it is possible to translate BDSs into propositional calculus and use SAT algorithms This article provides a theoretical study which paves the way for further developments and does not give stochastic is not yet at a validation stage on real gene regulation networks which would require for numerical simulation and benchmarking. Nevertheless links are done with modern SAT solvers which are particularly adopted tools in this framework

Biological systems, can be represented by a set of interacting elements (genes, proteins, metabolic enzymes, ...), whose states change over time. For genetic networks, the knowledge is regularly incomplete, updated, uncertain and sometimes

contradictory. It is therefore necessary to represent incomplete and revisable information. This representation has been studied in artificial intelligence since the late 1970s, especially by using non-monotonic logics and techniques derived from constraint programming. Notably, default logic (DL) [25, 7] as well as answer set programming (ASP) [20] have been used to study genetic networks.

In the context of biological systems, signaling pathways are specific interdependences between genes/proteins as responses of given cells to chemical signals or reactions of environment changes. The set of genes involved in these define gene regulation networks. These networks have been studied since the end of 1960s by using Boolean models derived from Boolean networks (BNs) [16, 38]. This supposes that a set of genes is associated to a set of variables together with their underlying Boolean functions. The evolution of the updates of these variables over discrete time constitutes a Boolean dynamical system (BDS). In this framework, it is considered that the expression of one gene modulates that of another gene trough an activation or inhibition action. The BDS approach allowed to find out feedback circuits theorems [10, 22, 26–31, 39], but also to model a wide range of processes of development biology [2, 9, 23], and physiology [6, 11, 21, 34] for instance.

We introduce representations in $\mathcal{H}$ for both BDSs *interaction graphs* and *asynchronous transition graphs* which allows us to exhibit new formal results. By using these representations, every Boolean network can be represented by a set of modal formulas of $\mathcal{H}$, and therefore by a set of *Kripke models* of $\mathcal{H}$. Most of the studies done on BDSs have focused on the analysis of both their stable configurations (or fixed points) and stable/unstable cycles/oscillations. This is the role of logic to state representation theorems between a language and the objects this language concerns. An important result of this paper is the proof that stable configurations and stable cycles are represented by sets of extensions: one *stable extension* for the former and a set of *ghost extensions* for the latter. To be practically usable, our formalism must also be able to compute extensions in an efficient way. The formalism introduced in this article uses a minimalist definition of the language of $\mathcal{H}$, but sufficient to represent BNs and their asynchronous dynamics. It is then possible to use very simple and efficient algorithms (notably stemming from the study of SAT). These algorithms allow us to distinguish among stable/unstable states. Moreover they can allow to enumerate all the states. This can be considered an important development regarding biological aspects.

Since this article deals with three areas, it is necessary to introduce the basics of these areas. This article is structured as follows: Section 2 reminds basic results about non-monotonic logic. Section 3 gives the main definitions and notations related to $\mathcal{H}$. Section 4 gives tracks to use SAT algorithms. Section 5 gives a representation of genetic networks into $\mathcal{H}$. Section 6 gives the definitions on BDS. Section 7 gives general fundamental results and applications in biology. Section 8, present a syntax and a semantic representations of BDSs in $\mathcal{H}$. Section 9, gives the theorems which prove that the asynchronous asymptotic behaviors such as

stable configurations and stable attractors, as well as unstable attractors, are properly captured. Section 10 gives a brief conclusion.

2 Non-monotonic logics and Default Logic

Gene regulation and signaling pathways can be a source of relevant studies regarding knowledge representation in artificial intelligence. Interactions in biological system appears as a form of causality. Therefore, the modeling of these systems is driven to use the logical inferences. The use of classical logic could be inadequate in this context because it cannot deal with inconsistencies. In addition, genetic networks are obtained by the analysis of long and expensive experiments, and basically, it is observed only a small part of the gene interactions. Finally, biologic knowledge is regularly incomplete, updated, uncertain and sometimes contradictory.

The representation of incomplete and revisable information has been studied in AI, especially by using of non-monotonic logics and techniques derived from constraint programming. Notably, *Default Logic*(DL) [25, 7] as well as *Answer Set Programming* (ASP) [20] have been used to study genetic networks. The representation of genetic networks with no circuits led to interesting results using an approach based on default logic [14, 13]. In this article we present a representation of Boolean genetic regulation networks using the non-monotonic modal logic called *hypothesis logic* ($\mathcal{H}$) [33, 36]. It was defined in 1993, after a first approach proposed by [3]. Some preliminary results on circuits using the Hypothesis Logic were already presented in [35].

Representing biological systems using a logical formalism may seem valid. For example the interaction of an entity i with j suggests a close relation with what is called in logic *material implication*, denoted as $\rightarrow$. However, such a representation in classical logic is not adapted because it drives often to inconsistencies. A way to manage this is to deal with a *non-monotonic formalism*. Monotonicity is a property of inference rules that never prunes the set of conclusions with the increasing of knowledge. Whereas this property is crucial in mathematics, it is largely questionable regarding reasoning with incomplete or contradictory information. This has led to the development of *non-monotonic logics* in artificial intelligence, including DL or ASP which is a more tractable restriction of DL. Default Logic concerns standard formulas of first order logic, to which contextual inference rules called defaults are added in order to deal with revisable information. A *default* is a local inference rule $d = \frac{A:B}{C}$, whose application specifically depends on the formulas A, B, C which 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 new information may invalidate previously triggered defaults. A first remark is that some default theories may have no extensions: this expresses a form of deep inconsistency which renders computation

more difficult. This possible lack of extensions in DL has been fully studied in the context of hypothesis logic. As shown in [36, 33], 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. A second remark is that DL only computes *stable extensions*, which can be a drawback regarding expressive power. Furthermore, such a type of extension cannot handle more than stable attractors of BDSs but we need to capture also unstable ones as well.

For biological systems, the main problem is to represent the *dynamics* and in particular the *cycles*. These cycles may represent real fundamental phenomena in living organisms such as the cell cycle [5, 19], circadian cycle [1, 32], or the cardiorespiratory rhythm [12]. This is the role of logic to state representation theorems between a language and the objects this language concerns. We show that $\mathcal{H}$ is a good candidate for such a language because it overtakes some of the limitations of default logic (and hence ASP also).

3 Hypothesis Logic

Hypothesis logic $\mathcal{H}$ is a bimodal logic [4] with two modal operators L and [H] [33, 36]. If f is a formula, the intuitive meaning of Lf is f *is proved/stated*. The dual H of [H] is defined as 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. This modal formula can also give translations of the Prolog clause and of the ASP rule: `C` :– A, not(B) .

The formalism used in this article uses a restricted definition of the language of $\mathcal{H}$, sufficient to represent BNs and boolean genetic networks. This restriction also allows to translate $\mathcal{H}$ into propositional calculus and to use SAT algorithms.

3.1 Syntax

The language of $\mathcal{H}$, denoted by $\mathscr{L}(\mathcal{H})$, is defined by the following inductive rules:

1. Any formula of propositional calculus (PC) is in $\mathscr{L}(\mathcal{H})$.
2. The set of atoms (or propositional variables) of $\mathscr{L}(\mathcal{H})$ is finite.
3. Whenever f and g are formulas of PC, Lf, $[\mathrm{H}]f$, Hf, $\neg\mathrm{L}f$, $\neg[\mathrm{H}]f$, $\neg\mathrm{H}f$ 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 [4] and [H] has those of the modal system K [4]. As a consequence, the inference rules and axiom schemata of $\mathcal{H}$ are:

[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.

1. All inference rules and axiom schemata of first-order logic.
2. (N[H]): $\vdash f \Rightarrow \vdash [\mathrm{H}]f$, the *necessitation rule* for [H].
3. (NL): $\vdash f \Rightarrow \vdash \mathrm{L}f$, the *necessitation rule* for L.
4. (K[H]): $\vdash [\mathrm{H}](f \to g) \to ([\mathrm{H}]f \to [\mathrm{H}]g)$, the *distribution axiom schema* for [H].
5. (KL): $\vdash \mathrm{L}(f \to g) \to (\mathrm{L}f \to \mathrm{L}g)$, the *distribution axiom schema* for L.
6. (TL): $\vdash \mathrm{L}f \to f$, 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*:

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

This very weak axiom is the basis 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: $\neg(\mathrm{L}f \wedge \mathrm{H}\neg f) \Leftrightarrow \mathrm{L}f \to \neg\mathrm{H}\neg f \Leftrightarrow \mathrm{H}\neg f \to \neg\mathrm{L}f$, 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

Kripke semantics [18] is defined for *normal modal logics* (*i.e.*, the logics which contain at least axiom (K)). We 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 PC. A world is then mapped to a logical interpretation. 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 [33], $\mathcal{H}$ 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 $\mathrm{RL} \subseteq \mathrm{R[H]}$ which corresponds to the link axiom. Proofs of completeness, correctness, and compactness for $\mathcal{H}$ are given in [33].

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) $\vdash \mathrm{L}f \to \mathrm{LL}f$. But, in S4 nested modalities of the same type collapse, so $\mathrm{LL}f = \mathrm{L}f$ and this addition makes us lose the notion of dynamics necessary for representing 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 LLp, and thus Lp by (TL), in all these accessible worlds, which, by induction would mean that p will be produced all the time. In a biological framework and for dynamical systems, this makes little sense. We encounter the same type of problem when adding axiom (5) $\vdash \mathrm{M}f \rightarrow \mathrm{LM}$.

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 any 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 proved. 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 [36, 33] 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 and the cycles of BDSs.

Definition 2. *Let E be an extension.*
• *E is* complete *if, for all $i \in V$,* $\mathrm{H}i \in E$ *or* $\mathrm{H}\neg i \in E$.
• *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, denoted by degree(E), is the number of free atoms that compose it.*
• *If the extension $E = \mathrm{Th}(\mathrm{F}(G) \cup \{\mathrm{H}y_k\}$, the* mirror *of E, denoted by* $\mathrm{mir}(E)$, *is the set* $\mathrm{Th}(\mathrm{F}(G) \cup \{H\neg y_k\})$.

4 Propositional algorithm

This section gives tracks to translate $\mathcal{H}$ into propositional calculus (PC) and to use SAT algorithms. Modal logics can be translated into first order logic by Skolemization methods. Then it is possible to use theorem provers based on the unification algorithm and the resolution principle. For example, it is possible to translate the modal operator L by a function $f\mathrm{L}$ of first order logic. But, by nesting the modalities, LL...p is translated by $f\mathrm{L}(f\mathrm{L}(...(p))$. The Herbrand universe (the set of ground formulas) is then infinite and therefore, the use of SAT algorithms is not appropriate.

On the contrary, with the restricted definition of $\mathcal{H}$ used here, there is no nesting of modalities (no LLf, HLf, ...). So, all modal formulas are of the form Lf, where f is a PC formula. If the set of propositional variables (atoms) is finite, we can then consider that the Herbrand universe is finite which opens the way for using SAT algorithms.

Let us recall some definitions of PC. A *literal* is an atom a, or the negation of an atom $\neg a$. A *disjunctive clause* (DC) is a disjunction of literals and, a *conjunctive clause* (CC) is a conjunction of literals. A *conjunctive normal form formula* (CNF formula) is a conjunction of one or more DCs. A *disjunctive normal form formula* (DNF formula) is a disjunction of one or more CCs. Every PC formula can be converted into an equivalent CNF formula and, also into an equivalent DNF formula.

Remember that for $\mathcal{H}$, the modal formulas are of the form Lp, Hp, or [H]p, where p is a PC formula. In the following a modal formula is *elementary* if p is a literal. Let F be a set a formulas of $\mathcal{H}$. We translate F into a set of PC formulas. This translation is done in the following way:

Translation:

1) Any modal formula Lp of F is translated into $\mathrm{L}p' = \mathrm{L}(c_1 \wedge ... \wedge c_n)$, where p' is the CNF of p. All c_i are therefore disjunctive clauses.

2) Any modal formula Hq of F is translated into $\mathrm{H}q' = \mathrm{H}(d_1 \vee ... \vee d_m)$ where q' is the DNF of q. All d_i are therefore conjunctive clauses.

3) The axiomatics of normal modal logics allows us to prove that:
- $\mathrm{L}(p_1 \wedge ... \wedge p_n) \equiv (\mathrm{L}p_1 \wedge ... \wedge \mathrm{L}p_n)$.
- $\mathrm{H}(q_1 \vee ... \vee q_n) \equiv (\mathrm{H}q_1 \vee ... \vee \mathrm{H}q_n)$

We can therefore replace any formula $\mathrm{L}(p_1 \wedge ... \wedge p_n)$ by a set $\{\mathrm{L}p_1, ..., Lp_n\}$ of disjunctive clauses and, any formula $\mathrm{H}(p_1 \wedge ... \wedge p_n)$ by a set $\{\mathrm{H}q_1, ..., \mathrm{H}q_n\}$ of conjonctive clauses.

5) We can then use a *renaming* which consists in replacing a PC formula by a new atom logically equivalent:

Any Lp_i (resp. Hq_j), such that p_i (resp. q_j) is not a literal, is replaced by a couple $\{\mathrm{L}x, x \equiv p_i\}$ (resp. by $\{\mathrm{H}y, y \equiv q_j\}$.

So F is replaced by a set of propositional formulas and elementary modal formulas. Note that it is also possible to translate each of these propositional formulas in its CNF form. ie a finite set of disjonctive clauses.

6) It remains to translate the set of elementary modal formulas into PC. This can be done by renaming each of these formulas by a new atom.

This translation is not enough for obtaining a SAT algorithm, because the axiomatic of $\mathcal{H}$ is not given. It remains to translate it into PC:
- The reflexivity axiom of $L : \mathrm{L}f \rightarrow f$
- The link axiom: $\neg(Lf \wedge H\neg)f$
- The distribution axiom for L: $\vdash \mathrm{L}(f \rightarrow g) \rightarrow (\mathrm{L}f \rightarrow \mathrm{L}g)$,
- The distribution axiom for [H] : $\vdash [\mathrm{H}]f \rightarrow g) \rightarrow ([\mathrm{H}]f \rightarrow [\mathrm{H}]g)$.

Since the set of CNF if is finite, each of these axioms can be translated into a set of modal propositional formulas where f and g take as values all the formulas of this set. Then we rename all the CNF formulas by new equivalent propositions. All modal formulas are then elementary, and it is possible to represent each of them by an atom.

If $\mathcal{T} = \{\mathrm{HY}, \mathrm{F}\}$ is an hypothesis theory, the computation of all the extensions of $\mathcal{T}$ is done by adding to the translation of F each translated subset of HY, and keeping only those among them that are the maximals consistent ones with F. This can be done by using a SAT algorithm. This solution exists, but it is, of course, not interesting because of the size of the translations.

Another possible solution would be to consider that each modality is a predicate of first order logic, and that axioms are first order formulas. For example the link axiom $Lf \rightarrow \neg\mathrm{H}\neg f$ is translated by $L(f) \rightarrow \neg H(\neg f)$, L and H being predicates. This translation is possible because there is no modality nesting ; moreover, the translation does not contain any functions. These formulas could be considered as global constraints which are used by the SAT algorithm when needed.

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

A *genetic network* represents interactions among genes or proteins in cell [2, 8, 9, 15, 17, 24]. From now on, let us consider that the entity at stake are proteins. In a context of modeling, 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 $\{0, 1\}$, which is legitimate when the focus is a qualitative modeling.

Genetic networks can be studied with the formalism of BNs and their underlying BDSs, defined in the following section. Here, in order to introduce our representation, it is sufficient 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: L$i \neq \neg\mathrm{L}\neg i$, $\neg\mathrm{L}i \neq \mathrm{L}\neg i$, 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. Let us consider the genetic network of a cell, and i ones of its protein. 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 to attempt 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 to attempt 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 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 for representing time, since it 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. In other words, $\mathcal{H}$ allows to formalize time, without having to add an additional level of representation.

The 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) L$i \rightarrow i$ *and* L$\neg i \rightarrow \neg i$ (i.e., *if i is produced (resp. destroyed), then i is present (resp. absent)).*
(2) $\neg$(L$i \wedge$ H$\neg i$) *and* $\neg$(L$\neg i \wedge$ Hi) *(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).*

Proof. PC and axioms of $\mathcal{H}$ are all what is needed.

(1) $\mathrm{L}i \rightarrow i$ and $\mathrm{L}\neg i \rightarrow \neg i$ are instances of axiom (T).

(2) $\neg(\mathrm{L}i \wedge \mathrm{H}\neg i)$ and $\neg(\mathrm{L}\neg i \wedge \mathrm{H}i)$ are instances of the linking axiom (LI).

(3) $Li \rightarrow i$ and $L\neg i \rightarrow \neg i$ are two instances of (1). In PC we have $(Li \rightarrow i) \rightarrow (\neg Li \vee i)$ and $(L\neg i \rightarrow \neg i) \rightarrow (\neg L\neg i \vee \neg i)$. Therefore we obtain $\neg Li \vee i$ and $\neg L\neg i \vee \neg i$ from which we derive, by resolution, $\neg \mathrm{L}i \vee \mathrm{L}\neg i \rightarrow \neg(\mathrm{L}i \wedge \mathrm{L}\neg i)$. □

6 Boolean dynamical systems

A finite BDS describes an 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 *dynamics* of such a network is modeled *via* both a function f, called the *transition function*, and an *updating mode* μ that defines how the elements of V are updated over time. More formally, $f : X \rightarrow X$ is such that $x = (x_1, ..., x_n) \mapsto f(x) = (f_1(x), ..., f_n(x))$, where each function $f_i : X \rightarrow \{0, 1\}$ is *a local transition function* that gives the evolution of i over time.

There is an uncountable number of updating modes[5]. Among them, the parallel and the fully asynchronous ones remain the most used [16, 38]. 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 [26, 29] which we will abbreviate by asynchronous dynamics for the sake of simplicity.

6.1 Asynchronous transition graphs

Let $X = \{0, 1\}^n$ be a configuration space and $f : X \rightarrow X$ a function that defines a BN. The *asynchronous dynamics* of f is given by its asynchronous transition graph (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:

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

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

[5] Uncountable because we can apply the Cantor's diagonal argument on the set of deterministic updating modes which are basically defined as infinite sequences of subsets of nodes of the network.

Note 3. By definition, we relate a unique ATG $\mathscr{G}(f)$ to a given function f. Moreover, by construction of ATGs, $\mathscr{G}(f) = \mathscr{G}(g)$ whenever f and g are equivalent (*i.e.*, $f(x) \equiv g(x)$ for every x), or again f and g have the same truth tables. A function, its truth tables and its ATG are therefore distinct representations of the same object, the asynchronous dynamical system.

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 the figures of this article, unless otherwise clearly specified recurring configurations, *i.e.*, configurations belonging to an attractor, are pictured in gray , and cycles are represented by bold transitions. Moreover, notice that the main difference between a stable cycle and an unstable one is that the first one represents a single orbit while the second one represents an infinity of orbits.

Example 1. Boolean positive and negative circuits of size 3:

Consider $V = \{1, 2, 3\}$, $x \in \{0, 1\}^3$ and two functions/BNs f and g such that $f(x_1, x_2, x_3) = (f_1(x), (f_2)(x), f_3(x)) = (\neg x_2, \neg x_3, x_1)$ and $g(x_1, x_2, x_3) = (g_1(x), (g_2)(x), g_3(x)) = (\neg x_3, x_1, x_2)$. From the definitions of f and g, it is possible to derive their related truth tables and ATGs, $\mathscr{G}(f)$ and $\mathscr{G}(g)$, pictured in Figure 1.

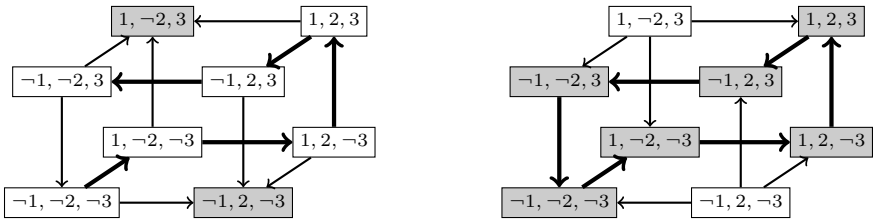
For these figures, the 8 rectangles are the vertices of the graph that represent the 2^3 possible assignments of $V = (1, 2, 3)$. A transition between 2 vertices corresponds to an arrow in these pictures. As the transition graph is asynchronous for each transition/arrow (x, y), x differs from y by a single component. Therefore, there are up to 3 transitions leaving each configuration.

Here, $\mathscr{G}(f)$ has two symmetric stable configurations, $(\neg 1, 2, \neg 3)$ and $(1, \neg 2, 3)$. These configurations are stable because no arrow comes out. The other six configurations induce a cycle, shown by the bold arrows. This cycle is unstable because it is possible to leave it, for example at vertex $(\neg 1, \neg 2, \neg 3)$.

$\mathscr{G}(g)$ has a cycle of length 6, shown in bold. This cycle is stable because it does not have any outward transition.

We will prove in Section 8 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.

The functions f and g can also be represented by *elementary circuits*, pictured in Figure 5. These ones are special cases of *interaction graphs*, defined below.



1	2	3	f_1	f_2	f_3	degree
0	0	0	1	1	0	2
0	0	1	1	0	0	2
0	1	0	0	1	0	0
0	1	1	0	0	0	2
1	0	0	1	1	1	2
1	0	1	1	0	1	0
1	1	0	0	1	1	2
1	1	1	0	0	1	2

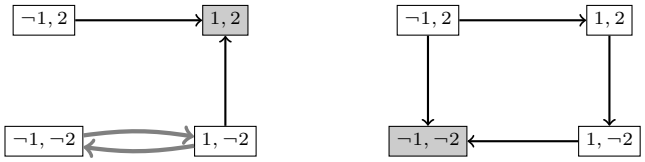
1	2	3	g_1	g_2	g_3	degree
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	1	0	1	0	1	3
0	1	1	0	0	1	1
1	0	0	1	1	0	1
1	0	1	0	1	0	3
1	1	0	1	1	1	1
1	1	1	0	1	1	1

Fig. 1. ATGs and truth tables with degree of freedom, of functions *(left)* f, and *(right)* g presented in Example 1.

Note 4. In order to draw the ATG of a function f, the easiest way is to use its truth table. For a line $l = (x_1, ..x_n, f_1(x), ...f_n(x))$ of the table, $(x_1, .., x_n)$ is a vertex of the ATG and $(f_1(x), ...f_n(x))$ will allow to draw the arcs starting from this vertex: for any x_i of l such that $x_i \neq f_i(x)$ then the picture contains an arc that goes from the node $(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ to the node $(x_1, \ldots, x_{i-1}, f_i(x), x_{i+1}, \ldots, x_n)$.

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 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.



1	2	h_1	h_2	degree
0	0	1	0	1
0	1	1	1	1
1	0	0	1	2
1	1	1	1	0

1	2	k_1	k_2	degree
0	0	0	0	0
0	1	1	0	2
1	0	0	0	1
1	1	1	0	1

Fig. 2. ATG and truth table with degree of freedom of functions *(left)* h and *(right)* k presented in Example 2 and 3.

Example 4 (Boolean positive circuit of size 4). Consider the BDS of function $f2(x_1, x_2, x_3, x_4) = (\neg x_4, x_1, \neg x_2, x_3)$, pictured in Figure 3. This BDS admits two stable configurations $(1, 2\neg 3, \neg 4)$ and $(\neg 1, \neg 2, 3, 4)$ pictured in gray, and an unstable oscillation, whose arcs are pictured in bold gray.

Example 5 (Boolean negative circuit of size 4). Consider the BDS of function $g2(x_1, x_2, x_3, x_4) = (\neg x_4, x_1, x_2, x_3)$, pictured in Figure 4. This BDS admits one stable cycle of length 8, whose arcs are pictured in bold and one unstable cycle of length 8, whose arcs are pictured in bold gray.

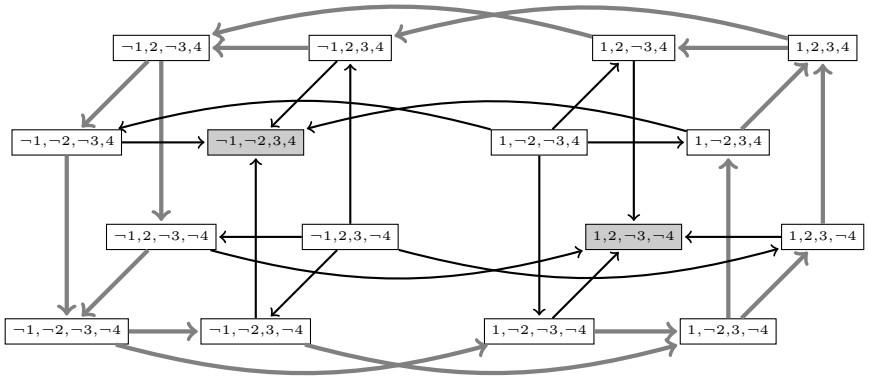


Fig. 3. ATG of the Boolean positive circuit of size 4 presented in Example 4.

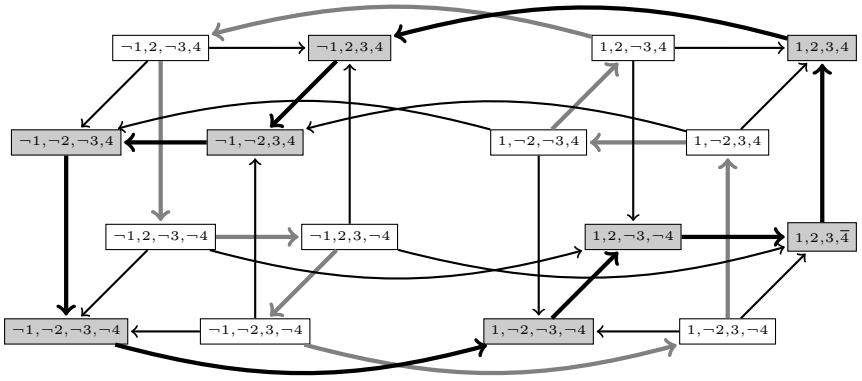


Fig. 4. ATG of the Boolean negative circuit of size 4 presented in Example 5.

6.2 Interaction graphs and circuits

An ATG is a very precise tool for studying the behavior of a function, but its size is exponential depending on the number of entities. Regarding practical applications, the information is often represented by more compact and more readable graphs of a different type, namely *interaction graphs* (IGs). This may be particularly the case for some biological data, which come from experiments that generally simply yield correlations among gene expressions.

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

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

Note 5. Consider the toy example where j has only one incoming arc from i. In this case, the effect of the regulation is obvious: if the arc is positive (resp. negative), j will take the value (resp. the opposite value) of i after one update. Remark that elementary circuits are regulated this way.

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

As mentioned above, an IG $G = (V, I)$ represents the existence of the interactions involved between the entities of V. Specifying the nature of these inter-

actions, and the conditions under which they occur effectively, leads to relate G to a function f which define a BN. Then, G is the IG of f and is then denoted by $G(f) = (V, I(f))$. This is done by assigning a local transition function f_i to every $i \in V$ so that $\forall j \in V, \exists x \in \{0,1\}^n,\ f_i(x) \neq f_i(\overline{x}^j) \iff (j, s \in \{+,-\}, i) \in I(f)$, where given $x = (x_1, \ldots, x_n)$, $\overline{x}^j = (x_1, \ldots, x_{j-1}, \neg x_j, x_{j+1}, \ldots, x_n)$. More precisely, by denoting the set of the variables of two configurations x and y having a different value by $\Delta(x, y) = \{i \in V \mid x_i \neq y_i\}$, $G(f) = (V, I(f))$ is such that:

- $(i, +, j) \in I_f$ if and only if there exist $x, y \in \{0,1\}^n$ with $\Delta(x, y) = \{i\}$ and $x_i = 0$ such that $f_j(x) = 0$ and $f_j(y) = 1$;
- $(i, -, j) \in I_f$ if and only if there exist $x, y \in \{0,1\}^n$ with $\Delta(x, y) = \{i\}$ and $x_i = 0$ such that $f_j(x) = 1$ and $f_j(y) = 0$.

This specification induces the minimality of $G(f)$ because each arc represents an effective interaction.

Example 6. Figure 5 pictures the IGs associated with the ATGs of the BDSs defined from f and g in Example 1. For these case of elementary circuits, a positive arc $\{i, +, j\}$ says that j takes the value of i and a *negative arc* $\{i, -, j\}$ says that j takes the value of $\neg i$. Consider the positive circuit associated with f by following the directions of its arcs:

- Starting from $x_1 = 1$, we get the infinite sequence: $(1, 3, \neg 2, 1, 3, \neg 2, \ldots)$.
- Starting from $x_1 = 0$, we get the infinite sequence: $(\neg 1, \neg 3, 2, \neg 1, \neg 3, 2, \ldots)$.

The first (resp. second) dynamical behavior highlights the stable configurations $1, 2, \neg 3$ (resp. $\neg 1, \neg 2, 3$) of f.

Consider the negative circuit associated with g:

- Starting from $x_1 = 1$ we get the infinite sequence $(\neg 1, \neg 2, \neg 3, 1, 2, 3, \neg 1, \neg 2, \neg 3, 1, 2, 3, \ldots)$.
- Starting from $x_1 = 0$, we get the infinite sequence $(1, 2, 3, \neg 1, \neg 2, \neg 3, 1, 2, 3, \neg 1, \neg 2, \neg 3, \ldots)$.

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

Figure 6 pictures the IGs of size 4, associated with functions presented in examples 4 and 5.

7 General fundamental results and applications in biology

By considering that BNs and their associated BDSs are good candidates for qualitatively modeling genetic networks (since established by the seminal papers [16, 38]), the presence of several attractors in their dynamical behaviors allows to

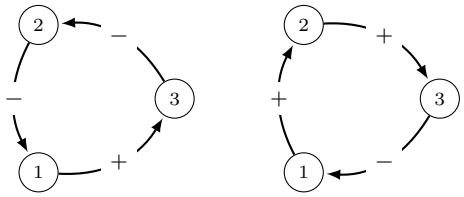


Fig. 5. *(left)* The IG (a positive circuit) associated with ATG $\mathscr{G}(f)$ and *(right)* the IG (a negative circuit) associated with ATG $\mathscr{G}(g)$, introduced in Example 1.

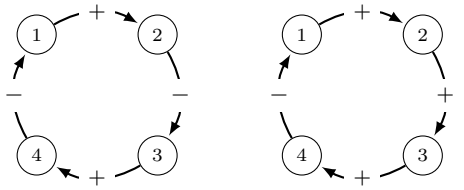


Fig. 6. *(left)* IG of $f2$ (Example 4) ; *(right)* IG of $g2$ (Example 5).

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 [37]. Another more tricky applications of BDSs in molecular systems biology concerns the floral morphogenesis of the plant *Arabidopsis thaliana* [23, 24]. Its dynamical behavior admits notably four stable configurations that correspond to the genetic expression patterns of the floral tissues, sepals, petal, stamens and carpels.

This model has also allowed to formally explain the role of the hormone gibberellin on the floral development [9]. 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 *interaction 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 [30, 31]. Moreover, in [39], basing himself on asynchronous DDS, Thomas conjectured that $G(f)$ of an asynchronous DDS f must contain a positive (resp. negative) circuit, for the latter to admit several stable configurations (resp. a non-trivial attractor such as a stable cycle or a stable oscillation). These two conjectures were proved to be true [26–29].

Furthermore, in [26], 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 $2n$).

Generalizations of this work to more complete combinations of circuits have been formally addressed in [22]. In [35], we obtained these results *via* the translation of BDSs into $\mathcal{H}$.

8 Representing BDS into $\mathcal{H}$

In [35], we studied in detail a translation of both positive and negative circuits into $\mathcal{H}$, which seemed 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 our translation to any asynchronous BDS. This translation does not use nesting of modalities and SAT algorithms can be used.

8.1 Syntax representation of BDS

Let's recall 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, remember 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:* $TR(f_i) = \{\mathrm{H}f_i(x) \rightarrow \mathrm{L}i,\ \mathrm{H}f_i(\neg x) \rightarrow \mathrm{L}\neg i\}$.
- *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* $TR(f) = \bigcup_{i=1}^{n} TR(f_i(x))$.

From the correspondence given in Note 3, this translation is equivalently the translation obtained for $\mathscr{G}(f)$ and the truth tables of f.

Example 7. Consider $V = \{1, 2, 3\}$, $X = \{0, 1\}^3$ and the function f of Example 1, defined as $x = (x_1, x_2, x_3)$ and $f(x) = (f_1(x), f_2(x), f_3(x)) = (\neg x_2, \neg x_3, x_1)$ whose ATG of is pictured in Figure 1. The functions f_1, f_2 and f_3 are translated into $\mathcal{H}$ by;
$TR(f_1)$=$\{\mathrm{H}2 \rightarrow \mathrm{L}\neg 1, \mathrm{H}\neg 2 \rightarrow \mathrm{L}1\}$, $TR(f_2)$=$\{\mathrm{H}3 \rightarrow \mathrm{L}\neg 2, \mathrm{H}\neg 3 \rightarrow \mathrm{L}2\}$, $TR(f_3)$= $\{\mathrm{H}1 \rightarrow \mathrm{L}3, \mathrm{H}\neg 1\neg \mathrm{L}3\}$.

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

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

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

Example 8. Consider function $g(x_1, x_2, x_3) = (\neg x_3, x_1, x_2)$ of Example 1. The translation in $\mathcal{H}$ leads to the following set of formulas: $F(G(g)) = \{H1 \rightarrow L2, H2 \rightarrow L3, H3 \rightarrow L\neg 1, H\neg 1 \rightarrow L\neg 2, H\neg 2 \rightarrow L\neg 3, H\neg 3 \rightarrow L1\}$.

We obtain the following 6 equivalent ghost extensions: $E1 = \{L2, L3\}$, $E2 = \{L\neg 1, L3\}$, $E3 = \{L\neg 1, L\neg 2\}$, $E4 = \{L\neg 2, L\neg 3\}$, $E5 = \{L1, L\neg 3\}$, $E6 = \{L1, L2\}$.

• $E1$, ..., $E6$ are extensions because they are consistent and it is impossible to add a hypothesis while remaining consistent. They are ghost extensions because in each of them there is a $\neg Hi$ (resp. $\neg H\neg i$) without L$\neg i$ (resp. Li).

• These extensions are of degree 1.

• In [35], we proved that there exists a permutation on the is that allows us to pass from $E1$ to $E2, ..., E6$ to $E1$. This permutation of 6 ghost extensions represents the stable cycle of g.

Moreover, there are also two sub-extensions, $E7 = \{1, \neg 2, 3\}$ and $E8 = \{\neg 1, 2, \neg 3\}$ that contain neither Li nor L$\neg i$. Hence all the is are free and their degree is 3. These extensions correspond to the configuration of g, of indegree 0 of the underlying BDS. They represent the *Garden of Eden* of g.

Example 9. Consider function k, such that $k(x_1, x_2) = (x_2, x_1 \wedge \neg x_1 \wedge x_2)$, whose ATG is pictured in Figure 2 right. Function k_1 is translated into $\mathcal{H}$ by the couple $TR(k_1) = \{H2 \rightarrow L\neg 1, H\neg 2 \rightarrow L1\}$ and function k_2 is translated by $TR(k_2) = \{H(1 \wedge \neg 1 \wedge 2) \rightarrow L2), H\neg(1 \wedge \neg 1 \wedge 2) \rightarrow L\neg 2\}$. 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: $TR(k) = \{H2 \rightarrow L\neg 1, H\neg 2 \rightarrow L1, H(1 \wedge \neg 1 \wedge 2) \rightarrow L2), H(\neg 1 \vee 1 \vee \neg 2) \rightarrow L\neg 2)\}$, which admits thee extensions:

• A stable extension $E1 = Th(TR(k) \cup \{H\neg 1, H\neg 2\}) = \{L\neg 1, L\neg 2\}$;

• Two ghost extensions of degree 1: $E2 = Th(TR(k) \cup \{H1, H\neg 1\}) = \{L\neg 2\}$, and $E3 = Th(TR(k) \cup \{H2\}) = \{L1\}$.

Note 6. Function k may appear naive, because $x_1 \wedge \neg x_1 \wedge x_2 = \bot$ ($\top$ is the logic formula $True$ and $\bot$ is $False$), which gives an equivalent translation $TR(h) = \{H2 \rightarrow L\neg 1, H\neg 2 \rightarrow L1, H\bot \rightarrow L2, H\top \rightarrow L\neg 2\}$. 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 $\mathcal{H}$ implicitly make the expected simplifications.

The following examples are interesting because they get out of simple cycles.

Example 10. Consider function $l1(x_1, x_2, x_3) = (\neg x_2 \vee x_3, x_1, \neg x_1)$ whose ATG is represented by the Figure 7 (left) and IG by Figure 8 (left). The translation of l_2 is $TR(f_1) =$

$\{\mathrm{H}(\neg 2 \vee 3) \rightarrow \mathrm{L}1, \mathrm{H}(2 \wedge \neg 3) \rightarrow \mathrm{L}\neg 1,$
$\mathrm{H}1 \rightarrow \mathrm{L}2, \mathrm{H}\neg 1 \rightarrow \mathrm{L}\neg 2,$
$\mathrm{H}1 \rightarrow \mathrm{L}\neg 3\}, \mathrm{H}\neg 1 \rightarrow \mathrm{L}3\}.$

We obtain 4 ghost extensions of degree 1:

$E1 = \{\mathrm{L}\neg 2, \mathrm{L}3\}$, $E2 = \{\mathrm{L}1, \mathrm{L}\neg 3\}$, $E4 = \{\mathrm{L}2, \mathrm{L}\neg 3\}$, $E7 = \{\mathrm{L}1, \mathrm{L}2\}$.

and 4 sub-extensions of degree 2:

$E3 = \{\mathrm{L}\neg 2\}$, $E4 = \{\mathrm{L}\neg 1\}$, $E5 = \{\mathrm{L}3\}$, $E8 = \{\mathrm{L}1\}$

In Figure 7 (left), the extensions of degree 1, correspond to the gray configurations. We notice that there is no stable cycle.

Example 11. Consider function $l2(x_1, x_2, x_3) = (x_2 \wedge \neg x_3, x_1, \neg x_1)$. Its ATG is represented by the Figure 7 (right) and its IG by Figure 8 (right). The translation of l_2 is $TR(f_1) =$

$\{\mathrm{H}(2 \wedge \neg 3) \rightarrow \mathrm{L}1, \mathrm{H}(\neg(2 \vee \neg 3)) \rightarrow \mathrm{L}\neg 1,$
$\mathrm{H}1 \rightarrow \mathrm{L}2, \mathrm{H}\neg 1 \rightarrow \mathrm{L}\neg 2,$
$\mathrm{H}1 \rightarrow \mathrm{L}\neg 3, \mathrm{H}\neg 1 \rightarrow \mathrm{L}3\}.$

We obtain 2 stable extensions :

$E2 = \{\mathrm{L}\neg 1, \mathrm{L}\neg 2, \mathrm{L}3\}$, $E7 = \{\mathrm{L}1, \mathrm{L}2, \mathrm{L}\neg 3\}$

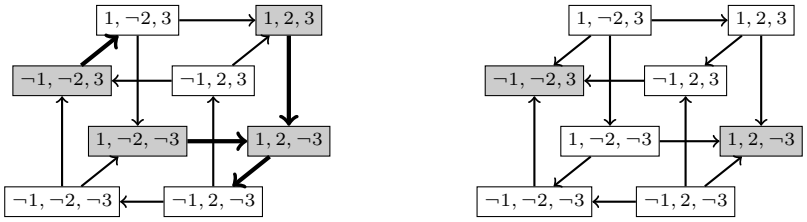
From these results, we retrieve the theorems about negative and positive double-cycles presented in [22] which gives a promising strong correspondence between $\mathcal{H}$ and BDS. This correspondance will be formally studied in the rest of this article.

8.2 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* [18] presented in Section 3.2. In order to obtain these morphisms, we give an 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 $\mathscr{G}(f) = (X, T(f))$. Remember 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 its 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.



1	2	3	$l1_1$	$l1_2$	$l1_3$	ext.	deg
0	0	0	1	0	1	E1	2
0	0	1	1	0	1	E2	1
0	1	0	0	0	1	E3	2
0	1	1	1	0	1	E4	2
1	0	0	1	1	0	E5	1
1	0	1	1	1	0	E6	2
1	1	0	0	1	0	E7	1
1	1	1	1	1	0	E8	1

1	2	3	$l2_1$	$l2_2$	$l2_3$	ext.	deg
0	0	0	0	0	1	E1	1
0	0	1	0	0	1	E2	0
0	1	0	1	0	1	E3	3
0	1	1	0	0	1	E4	1
1	0	0	0	1	0	E5	2
1	0	1	0	1	0	E6	3
1	1	0	1	1	0	E7	0
1	1	1	0	1	0	E8	2

Fig. 7. *(left)* ATG and truth tables of function $l1$, and *(right)* of function $l2$, presented in Example 10 and Example 11. The gray rectangles are representing extensions of degree 1 (left) and stable configurations (right).

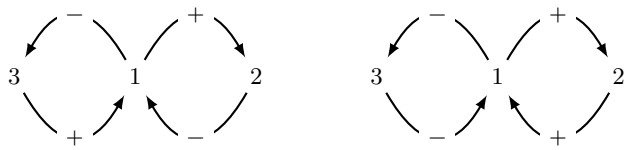


Fig. 8. (left) IG of $l1$, Example 10 and (right) IG of $l2$, Example 11

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 which does 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 mod-

els. 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 12. Figure 9 depicts the projection of function f given in Example 1, and its associated Krypke model $K(f)$.We remark that in order to get $K(f)$ from f, it is enough to inject the modality L at the right places into the cube, according to the corresponding Kripke frame. The eight nodes of $K(f)$ are the worlds, and the arcs express the accessibility relation. The two nodes in gray $\{\mathrm{L}1, \mathrm{L}\neg 2, \mathrm{L}3\}$ and $\{\mathrm{L}\neg 1, \mathrm{L}2, \mathrm{L}\neg 3\}$ represent the two stable extensions from the translation of f. Their degree of freedom is 0. The other six nodes are sub-extensions since they are non-maximal. Their degree is 2. These six nodes form the unstable cycle of f, represented by bold transitions.

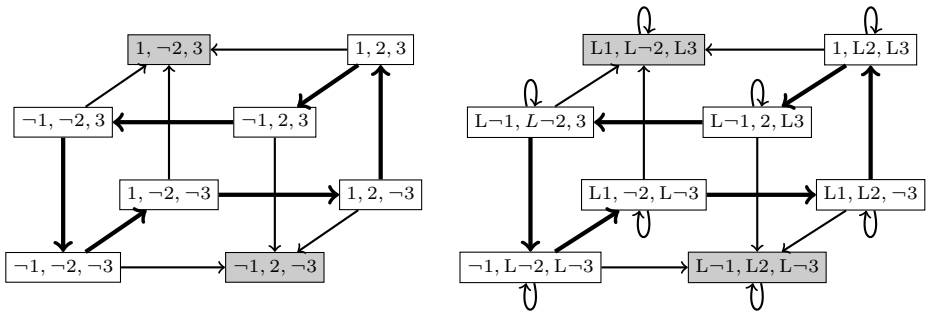


Fig. 9. *(left)* ATG of function f given in Example 1, and *(right)* its associated Kripke model.

Example 13. Figure 10 depicts the ATG of function g given in Example 1 and it's associated Kripke model. The six nodes in gray represent the ghost extensions of the translation of g. Their degree of freedom is 1 and correspond to the six configurations of the stable cycle of function g, represented by bod transitions. The other two nodes $\{1, \neg 2, 3\}$ and $\{\neg 1, 2, \neg 3\}$ are sub-extensions because they are non-maximal; their degree of freedom is 3. They represent the Garden of Eden of g.

Example 14. Figure 11 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, $\{1, 2\}$.

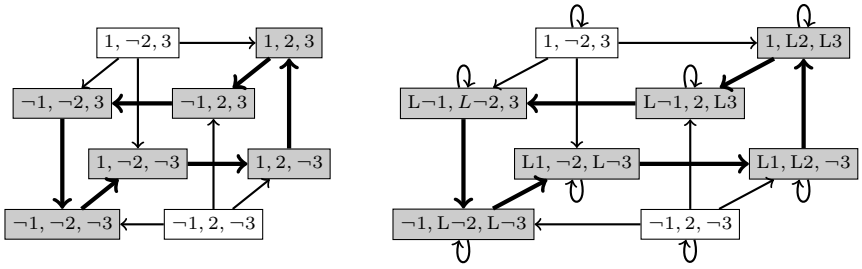


Fig. 10. *(left)* ATG model of function g given in Example 1 and *(right)* its associated Kripke model.

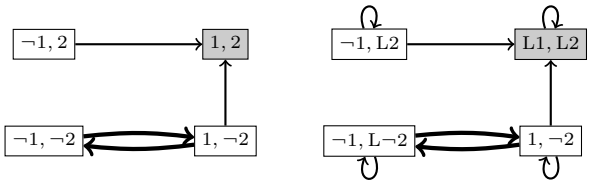


Fig. 11. *(left)* The ATG of h, *(rigth)* the corresponding Kripke model.

9 Results

The existence of a morphism from hypothesis theories to ATGs, allows to prove the following theoretical results. Therefore, there is a strong link between configurations of a BDS and extension sets of the $\mathcal{H}$ representation of this BDS.

The generalization of the notion of degree of freedom of extensions allows us 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 $i \in \{i_1, \ldots, i_k\}$, 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 $i \in \{i_1, \ldots, i_k\}$, there is a world accessible from w, and distinct from w, that contains the opposite of i. Because

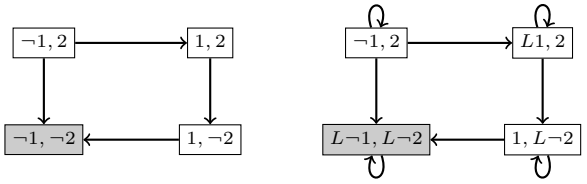


Fig. 12. *(left)* The ATG of k, *(rigth)* the corresponding Kripke model.

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 $x = \{x_1, \ldots, x_n\}$ is a stable configuration of $\mathscr{G}(f)$, then there exists an extension E of degree 0, issued from $TR(f)$, that contains $\{\mathrm{L}x_1, \ldots, \mathrm{L}x_n\}$.

[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 proved 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 $C = \{x_1, \ldots, x_k\}$ be a stable cycle of $\mathscr{G}(f)$, and $W = \{w_1, \ldots, w_\ell\}$ 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 [35]. They correspond to the results given in [26]. 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.

10 Conclusion

In [35], 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 be used 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.

Our study is only a first step toward a complete *logical* study of BDSs that, in the end, should allows us to clear both semantical and computational interesting properties of BDSs.

Acknowledgements. This work was primarily funded by our salaries as French State agents (SS being officially affiliated to Aix-Marseille Univ, CNRS, LIS, Marseille, France), and secondarily by Agence Nationale pour la Recherche (ANR) in the scope of the project ANR-18-CE40-0002 FANs (SS).

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. Dealy, S., Kauffman, S., Socolar, J.: Modeling pathways of differentiation in genetic regulatory networks with Boolean networks. Complexity **11**, 52–60 (2005)
7. Delgrande, J.P., Schaub, T.: Expressing default logic variants in default logic. Journal of Logic and Computation **15**, 593–621 (2005)
8. Demongeot, J., Elena, A., Noual, M., Sené, S., Thuderoz, F.: "Immunetworks", intersecting circuits and dynamics. Journal of Theoretical Biology **280**, 19–33 (2011)
9. 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)
10. Demongeot, J., Noual, M., Sené, S.: Combinatorics of Boolean automata circuits dynamics. Discrete Applied Mathematics **160**, 398–415 (2012)
11. Demongeot, J., Sené, S.: About block-parallel Boolean networks: a position paper. Natural Computing **19**, 5–13 (2020)
12. 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)

13. 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)
14. 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)
15. 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)
16. Kauffman, S.A.: Metabolic stability and epigenesis in randomly constructed nets. Journal of Theoretical Biology **22**, 437–467 (1969)
17. Kauffman, S.A., Peterson, C., Samuelsson, B., Troein, C.: Random Boolean network models and the yeast transcriptional network. PNAS **100**, 14796–14799 (2003)
18. Kripke, S.A.: Semantical analysis of modal logic I Normal modal propositional calculi. Mathematical Logic Quarterly **9**, 67–96 (1963)
19. Li, F., Long, T., Lu, Y., Ouyang, Q., Tang, C.: The yeast cell-cycle network is robustly designed. PNAS **101**, 4781–4786 (2004)
20. Lifschitz, V.: Action languages, answer sets, and planning. In: The Logic Programming Paradigm: A 25-Year Perspective, pp. 357–373. Springer (1999)
21. Liquitaya-Montiel, A.J., Mendoza, L.: Dynamical analysis of the regulatory network controlling natural killer cells differentiation. Frontiers in Physiology **9**, 1029 (2018)
22. 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)
23. 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)
24. Mendoza, L., Thieffry, D., Alvarez-Buylla, E.R.: Genetic control of flower morphogenesis in *Arabidopsis thaliana*. Bioinformatics **15**, 593–606 (1999)
25. Reiter, R.: A logic for default reasoning. Artificial Intelligence **13**, 81–132 (1980)
26. Remy, E., Mossé, B., Chaouyia, C., Thieffry, D.: A description of dynamical graphs associated to elementary regulatory circuits. Bioinformatics **19**, ii172–ii178 (2003)
27. 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)
28. Richard, A.: Negative circuits and sustained oscillations in asynchronous automata networks. Advances in Applied Mathematics **44**, 378–392 (2010)
29. Richard, A., Comet, J.P.: Necessary conditions for multistationarity in discrete dynamical systems. Discrete Applied Mathematics **155**, 2403–2413 (2007)
30. Robert, F.: Itérations sur des ensembles finis et automates cellulaires contractants. Linear Algebra and its Applications **29**, 393–412 (1980)
31. Robert, F.: Discrete Iterations: A Metric Study. Springer (1986)
32. Roenneberg, T., Merrow, M.: The network of time: understanding the molecular circadian system. Current Biology **13**, R198–R207 (2003)
33. Schwind, C., Siegel, P.: A modal logic for hypothesis theory. Fundamenta Informaticae **21**, 89–101 (1994)
34. Shmulevich, I., Dougherty, E.R., Zhang, W.: From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. Proceedings of the IEEE **90**, 1778–1792 (2002)

35. 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)
36. Siegel, P., Schwind, C.: Modal logic based theory for non-monotonic reasonning. Journal of Applied Non-classical Logic **3**, 73–92 (1993)
37. Thieffry, D., Thomas, R.: Dynamical behaviour of biological regulatory networks – II. Immunity control in bacteriophage Lambda. Bulletin of Mathematical Biology **57**, 277–297 (1995)
38. Thomas, R.: Boolean formalization of genetic control circuits. Journal of Theoretical Biology **42**, 563–585 (1973)
39. 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)