Natural Computing (2023) 22:431–451
https://doi.org/10.1007/s11047-023-09945-2

Linear cuts in Boolean networks

Aurélien Naldi[1] · Adrien Richard[2] · Elisa Tonello[3]

Accepted: 24 May 2023 / Published online: 4 July 2023

Abstract
Boolean networks are popular tools for the exploration of qualitative dynamical properties of biological systems. Several dynamical interpretations have been proposed based on the same logical structure that captures the interactions between Boolean components. They reproduce, in different degrees, the behaviours emerging in more quantitative models. In particular, regulatory conflicts can prevent the standard asynchronous dynamics from reproducing some trajectories that might be expected upon inspection of more detailed models. We introduce and study the class of networks with linear cuts, where linear components—intermediates with a single regulator and a single target—eliminate the aforementioned regulatory conflicts. The interaction graph of a Boolean network admits a linear cut when a linear component occurs in each cycle and in each path from components with multiple targets to components with multiple regulators. Under this structural condition the attractors are in one-to-one correspondence with the minimal trap spaces, and the reachability of attractors can also be easily characterized. Linear cuts provide the base for a new interpretation of the Boolean semantics that captures all behaviours of multi-valued refinements with regulatory thresholds that are uniquely defined for each interaction, and contribute a new approach for the investigation of behaviour of logical models.

Keywords Boolean networks · Updating semantics · Reachability · Attractors

Mathematics Subject Classification 94C99 · 92B05 · 06E30 · 68Q10 · 37B15

1 Introduction

Boolean networks are a class of non-deterministic discrete event systems used as qualitative dynamical models of biological processes. The study of complex biological processes leads to two types of results: insight about the internal molecular mechanisms, and observation of their state over time and different external stimulations. While the changes of state emerge from the internal mechanisms, they can not be directly compared. The integration of mechanistic knowledge into dynamical models enables to contrast the behaviour emerging from the model with the experimental observations. Such models are valuable tools to identify inconsistencies, evaluate hypothesis and prioritize their experimental validation. Starting with a known initial condition, the model can be used to predict the reachability and stability of a target phenotype, which corresponds to properties of the reachable states of the model. The lack of precise information on the initial conditions and kinetic parameters impedes the construction of comprehensive quantitative models without performing time-consuming exploration of parameters. Boolean and more generally qualitative models have been proposed to cope with this lack of quantitative knowledge (Kauffman 1969; Thomas 1973). These discrete models are well suited to build large comprehensive models based on incomplete knowledge. They are also amenable to formal analysis, in particular for the identification of attractors (Naldi et al. 2007; Dubrova and Teslenko 2011; Klarner et al. 2014). Multi-valued networks can be used to account for components for which a higher activity level (denoting for

✉ Elisa Tonello
elisa.tonello@fu-berlin.de

Aurélien Naldi
aurelien.naldi@inria.fr

Adrien Richard
adrien.richard@cnrs.fr

1 Lifeware Group, Inria Saclay-Ile de France, Palaiseau, France

2 I3S, Université Côte d'Azur - CNRS, Sophia Antipolis, France

3 Department of Mathematics and Computer Science, Freie Universität, Berlin, Germany

example a higher concentration or a stronger activation) can lead to different effects (new targets, stronger or different effect). Most large networks lack this level of detail and consider only Boolean components (sometimes a few selected multi-valued components). In practice, the coarse-grained predictions obtained with these models are sufficient to reproduce relevant behaviours in a wide range of biological applications (e.g. Sizek et al. 2019; Béal et al. 2021; Bonzanni et al. 2013; Cohen et al. 2015; Collombet et al. 2017).

The analysis of these models often aims initially at the identification of attractors (fixed points or stable oscillations) and reachability properties, which are computationally hard problems in the classical asynchronous semantics. Modelers can attempt to simplify the analysis by first considering *trap spaces*, stable subspaces that can be efficiently identified using constraint-solving approaches (Klarner et al. 2015; Chevalier et al. 2019; Trinh et al. 2022). Trap spaces provide a crude approximation of some of the attractors, but may not capture all of them. In addition, they can be used to rule out some reachability properties, since all states outside of the smallest trap space including the initial state are not reachable. On the other hand, reachability analysis inside a given trap space remains hard to solve. These questions are much easier to tackle using the recently proposed *most permissive* semantics (Paulevé et al. 2020), an over-approximation of the asynchronous semantics which lifts competition between concurrent events by introducing intermediate states representing the inherent uncertainty of Boolean networks. This approach formally accounts for the reachability properties of all possible refinements and uncovers missing realistic behaviours that are not captured by the asynchronous semantics. This results in very good computational properties, with all attractors being trap spaces and reachability analysis being polynomial. On the other hand, the most permissive semantics can also introduce non-monotonic behaviours, which may contradict the original intent of the model and could be considered as artefacts.

In this work, we propose an alternative semantics based on structural properties underlying the competition between components. We study constraints on the order of events in the asynchronous and most permissive semantics, specifically those related to the existence of maximal geodesics. We introduce the class of Boolean networks that admit a *linear cut*, that is, networks in which every cycle and every path from a component with multiple targets to a component with multiple regulators contains at least one linear component (a component with in- and out-degree equal to one). In essence, these linear components can be used to relax competitions between other components in the network. We show that for all initial states with stable linear components (*canonical states*), maximal geodesics of the most permissive semantics exist in the asynchronous dynamics. We prove two main consequences of this observation: (1) minimal trap spaces provide a precise (but not exact) characterization of all attractors; and (2) given a canonical initial condition, all trap spaces (hence all attractors) included in the smallest trap space containing the initial state are reachable. The characterization of reachability from other states and subspaces remains a hard problem. While many realistic Boolean networks do not satisfy the required topological properties, we show that one can always construct an extended network which does, by adding intermediate components on competing interactions. We use this extension to define a new semantics which is an over-approximation of the classical asynchronous semantics and an under-approximation of the most-permissive semantics. In Boolean networks of biological systems, interactions are often abstract representations summarizing multiple intermediate steps, hence networks resulting from the addition of explicit intermediates can presumably be considered as valid candidate models. In these cases, the extended semantics takes advantage of some key computational properties of the most permissive semantics with a higher confidence in the interpretability of the results.

In Sect. 2, we present classical concepts and formal notation used in this work. In Sect. 3, we introduce *implicant maps* as a tool to study constraints between transitions in asynchronous and permissive trajectories. In Sect. 4, we define the topological class of *L-cuttable* Boolean networks and derive some of their key dynamical properties, in particular the one-to-one correspondence between minimal trap spaces and attractors. In Sect. 5 we show that extended networks, accounting for realistic delay effects, can be used to take advantage of the dynamical properties of cuttable networks to investigate any Boolean network, and to recreate behaviours of a class of monotonic multi-valued refinements. Finally, we discuss how the semantics of linearly extended networks relate to the asynchronous and permissive semantics, and their potential practical application to the exploration and validation of biological models.

2 Background

In this section we introduce notations and definitions used throughout the paper. The symbol $\mathbb{B}$ will denote the set $\{0, 1\}$. Given a set A, we will write $\mathcal{P}(A)$ for the power set of A.

A *Boolean network* is defined by a pair $M = (V, f)$, where $V = \{1, \ldots, n\}$ is called the set of variables or components of the Boolean network, and f is an endomorphism of $\mathbb{B}^V$.

The set $\mathbb{B}^V$ will be called the set of states of the Boolean network, sometimes called state space. Any pair of states x and y delimit a *subspace* $[x, y]$ defined as the subset of states $\{z \in \mathbb{B}^V \mid z_i = x_i = y_i \text{ for all } i \in V \text{ s.t. } x_i = y_i\}$. We will denote subspaces also as elements of $\{0, 1, \bigstar\}^V$, so that a state x belongs to a subspace $t \in \{0, 1, \bigstar\}^V$ if for all variables i we have either $x_i = t_i$ or $t_i = \bigstar$. That is, we use $\bigstar$ to represent free variables. Note that states are subspaces without free variables and that a subspace with k free variables contains 2^k different states.

Given a Boolean network (V, f) and a component $i \in V$, we call $f_i : \mathbb{B}^V \to \mathbb{B}$ the Boolean function associated to the component i. An *implicant* of f_i is a subspace t such that $f_i(x) = 1$ for all states $x \in t$. An implicant is *prime* if it is not contained in any larger implicant (i.e. if it has a minimal set of fixed variables).

For a subset A of $\mathbb{B}^V$, $[A]$ will denote the minimal subspace containing A, and $\Delta(A)$ the set of free variables of $[A]$ (that is, $\Delta(A) = \{i \in V \mid \exists x, y \in A \text{ s.t. } x_i \neq y_i\}$). If A consists of two states x and y, we will write $\Delta(x, y)$ for $\Delta(A)$. We extend the notation $\Delta(x, y)$ to apply to elements x, y of $\{0, 1, \bigstar\}^V$ as follows: $i \in \Delta(x, y)$ if and only if $x_i \neq \bigstar$, $y_i \neq \bigstar$ and $x_i \neq y_i$.

Given a state $x \in \mathbb{B}^V$ and a set of components $I \subseteq V$, we define the state $\bar{x}^I$ by $\bar{x}^I_i \neq x_i$ for all $i \in I$ and $\bar{x}^I_j = x_j$ for all $j \in V \backslash I$. By convention, $\bar{x}^i = \bar{x}^{\{i\}}$.

For a Boolean network (V, f), we say that $i \in V$ is a *regulator* of $j \in V$ if there exists $x \in \mathbb{B}^V$ such that $f_j(x) \neq f_j(\bar{x}^i)$. In this case, component j is called a *target* for i. We will use the notations $R, T : V \to \mathcal{P}(V)$ to denote the maps that give the set of regulators and targets of components, respectively.

The *interaction graph* of a Boolean network (V, f) summarises the regulations between components. It is the graph with set of vertices V and set of edges defined by $\{(i,j) \in V \mid j \in T(i)\}$. The edges of the interaction graph are also called interactions of the network.

The dynamical behaviour of a Boolean network (V, f) is encoded in transitions between states. These transitions are defined by the Boolean rules f and an updating semantic, which can be deterministic (each state has a single successor) or non-deterministic (each state can have multiple successors defining alternative dynamical trajectories). The deterministic synchronous updating was first proposed by Kauffman (1969). In this work, we focus on the non-deterministic asynchronous updating, introduced by Thomas (1973). As the name suggests, the synchronous updating assumes that all possible changes always happen at the same time, while the asynchronous updating assumes that all changes happen separately. In the generalized asynchronous updating, changes can happen either at the same time or separately: it contains all transitions from the synchronous and asynchronous updatings, as well as all other transitions where a subset of components are updated. More in detail, for a Boolean network (V, f), given two distinct states x, y (i.e. $\Delta(x, y) \neq \emptyset$), there exists a transition from x to y

- in the synchronous dynamics, if and only if $y = f(x)$,
- in the asynchronous dynamics, if and only if $y = \bar{x}^i$ with $i \in \Delta(x, f(x))$,
- in the generalized asynchronous dynamics, if and only if $\Delta(x, y) \subseteq \Delta(x, f(x))$.

Note that each state has at most one successor in the synchronous updating, at most n successors in the asynchronous updating and up to $2^n - 1$ successors in the generalized asynchronous case.

Other updatings have been proposed, in particular the bloc-sequential updating (deterministic, see Robert 1986) and the use of priority classes (non-deterministic, see Fauré et al. 2006). In addition, one can define stochastic dynamics by adding transition probabilities to non-deterministic updatings. A trajectory from a state x to a state y in any of these updatings implies the existence of a trajectory from x to y in the generalized asynchronous dynamics. By definition, all transitions in the synchronous, asynchronous and priority updatings are also transitions in the generalized asynchronous dynamics. Individual bloc-sequential transitions may not correspond to transitions in the generalized asynchronous dynamics; however, equivalent trajectories always exist. In summary, the reachability properties of the generalized asynchronous dynamics provide an over-approximation of the reachability properties in all other classical updatings. More precisely, the set of reachable states in the generalized asynchronous dynamics contains those of other updatings.

To discuss reachability properties and asymptotic dynamics, we will need some additional nomenclatures. For a path or trajectory P in the asynchronous dynamics given by the sequence of states $x^0, \ldots, x^l$, we call *direction sequence* of the path P the sequence $i_0, \ldots, i_{l-1}$ of the directions of the edges in the path, and we call l the *length* of the path. In other words, the sequences satisfy $x^k_{i_k} \neq x^{k+1}_{i_k}$ for $i = 0, \ldots, l-1$. If the direction sequence contains no repetition, we say that P is a *geodesic*. For convenience, we will call a geodesic in asynchronous dynamics an *asynchronous geodesic*.

A *fixed point* (also called stable state or steady state), is a state x such that $f(x) = x$. Given a fixed point x, we have $\Delta(x, f(x)) = \emptyset$, and this state has no successor in any updating.

A *trap space* (also called stable motif), is a subspace t such that for each $x \in t$, $f(x) \in t$ (equivalently, for all $i \in V$, either $t_i = \bigstar$ or $f_i(x) = t_i$ for all $x \in t$). One can think of trap spaces as partial fixed points. If a state belongs to a trap space, then all its successors in any updating also belong to this trap space. We call a trap space minimal if it is not a superset of any other trap space. Note that the overlap of two trap spaces is also a trap space and that there is a unique minimal trap space containing a given state x.

Given $K \subseteq V$, a K-trap space is a subspace t such that for all $i \in K$, either $t_i = \bigstar$ or $f_i(x) = t_i$ for all $x \in t$. In other words, a K-trap space is closed for the dynamics when only the variables in K are considered. V-trap spaces are therefore standard trap spaces.

A *trap set* is a subset $S \subseteq \mathbb{B}^n$ of the state space that is closed for the dynamics, meaning that $f(x) \in S$ for all states $x \in S$. A trap space is therefore a trap set that is also a subspace. An *attractor* is an inclusion-minimal trap set. It can consist of an isolated state (it is then a fixed point), or of multiple states; in the latter case it is called a *cyclic* or *complex* attractor. Note that trap sets and attractors may depend on the updating semantics, while fixed points and trap spaces are structural properties of the network itself. Each trap space is also a trap set and contains at least one attractor for any updating; the number of minimal trap spaces is thus a lower bound for the number of attractors.

Example 1 The Boolean network $f(x_1, x_2) = (x_2, x_1)$ has two fixed points, 00 and 11. These are also the only attractors of the asynchronous dynamics. The state $f(01) = 10$ is reachable from 01 in the synchronous and generalized asynchronous dynamics, but not in the asynchronous dynamics. This network has three trap spaces: the state space $\mathbb{B}^2$ (which is not minimal) and the two fixed points. The set $\{01, 10\}$ is a trap set and an attractor of the synchronous dynamics (a cyclic attractor), but not of the asynchronous and generalized asynchronous.

In addition to the classical updating semantics, the *most permissive* (MP) semantics has recently been proposed to account for trajectories of multi-valued or continuous refinements which are not captured by the generalized asynchronous dynamics (Paulevé et al. 2020). This semantics can be defined by introducing intermediate states ("increasing" or "decreasing") representing uncertainty during the transitions from regular Boolean states: when a component is in an intermediate state, its target can behave as if it were in either of the classical Boolean state. The reachability of Boolean states in the most permissive semantics can be formulated as follows (see Paulevé et al. 2020, Supplementary Information):

Given x and $y \in \mathbb{B}^n$, y is reachable from x in the most permissive semantics if and only if there exist $K \subseteq V$ such that the smallest K-trap space t containing x also contains y, and for all $i \in K$ there exists $z \in t$ such that $f_i(z) = y_i$.

To each transition in the general asynchronous dynamics (which include all transitions of the synchronous and asynchronous dynamics) corresponds at least one path in the most permissive semantics: given $K \subseteq \Delta(x, f(x))$, the smallest K-trap space containing x contains $\bar{x}^K$.

From a computational perspective, while the most permissive semantics increases the cost of explicit simulations due to its large number of trajectories, it also enables efficient analytical methods for the identification of attractors and reachability properties.

3 Partial orders in asynchronous and permissive trajectories

Here we investigate structural conditions guaranteeing that all reachable states in the most permissive semantics are also reachable in the asynchronous dynamics. For this, we define *permissive trajectories*, inspired by the definition of most permissive semantics. We will then use implicants associated to the Boolean function and their differences with the initial state to identify partial orders enabling permissive geodesics. The partial orders that satisfy additional constraints correspond to geodesics in the classical asynchronous dynamics.

Definition 1 A *permissive trajectory* is a sequence of states $x^0, x^1, \ldots, x^l$ such that for any $k < l$ there is a component i such that $\Delta(x^k, x^{k+1}) = \{i\}$ and the smallest subspace containing all states $\{x^0, \ldots, x^k\}$ contains at least one state y such that $f_i(y) = x_i^{k+1}$. By extension, a *permissive geodesic* is a permissive trajectory where each component is used at most once. We say that the geodesic is *maximal* if it cannot be extended to a permissive geodesic of greater length.

An asynchronous trajectory is clearly also a permissive trajectory. We further observe that, if there is a transition from a state x to a state y in the generalized asynchronous dynamics, then there is a permissive geodesic from x to y. By extension, if there is a generalized asynchronous trajectory from x to y, then there is also a permissive trajectory from x to y (but not necessarily a geodesic). It can also be shown that there is a permissive trajectory from x to y if and only if y is reachable from x in the most permissive semantics.

Proposition 1 *Given any permissive trajectory from x to y, there exists a permissive trajectory from x to y of length at most 2n.*

This property corresponds to Lemma 1 in Paulevé et al. (2020) (Supplementary Information). Note that we get a bound of $2n$ steps here instead of the $3n$ bound in Paulevé et al. (2020) as the transitions from transitory states to regular Boolean states are implicit in the definition of the permissive trajectories.

Proposition 2 *Let x be a state and let y be such that [x, y] is the minimal trap space containing x. Then all maximal permissive geodesics starting in x end in y.*

Proof Consider a maximal permissive geodesic P from x to a state z. Since P is maximal, $f_i(w) = x_i$ for all $w \in [x, z]$ and $i \notin \Delta(x, z)$, that is, $[x, z]$ is a trap space, hence it contains $[x, y]$, so $\Delta(x, y) \subseteq \Delta(x, z)$. Suppose that $\Delta(x, z) \setminus \Delta(x, y)$ is not empty, and take the first variable $i \in \Delta(x, z) \setminus \Delta(x, y)$ that changes along P. Then $f_i(w) \neq x_i$ for some $w \in [x, y]$, which contradicts the fact that $[x, y]$ is a trap space. Hence $\Delta(x, y) = \Delta(x, z)$, which concludes. □

Example 2 Consider the Boolean network with $V = \{1, 2\}$ and $f(x_1, x_2) = (0, x_1)$. The trajectory 10, 11, 01 is an asynchronous geodesic from 10 to 01. The sequence of states 10, 00, 01 is not an asynchronous trajectory, it is however a permissive geodesic, since the state 10 satisfies $f_1(10) = 0, f_2(10) = 1$.

Example 3 The Boolean network with $V = \{1, 2, 3\}$, $f(x_1, x_2, x_3) = (1, x_1, x_2 \wedge \neg x_1)$ (see Fig. 1a) admits the permissive geodesic 000, 100, 110, 111, but no asynchronous trajectory from 000 to 111.

We are interested in studying reachability from a given initial condition x. In particular we are interested in determining whether a target state is reachable from x by looking at the implicants defining the network f. To this end, we introduce *implicant maps*, which encode possible choices of implicants for each component involved in a trajectory. Note that the number of potential implicant maps increases exponentially with the number of implicants of the components. Furthermore, the number of possible implicants of a given component increases exponentially with the number of its regulators. In practice, most components in biological networks have a small number of regulators and these problems remain tractable for example using constraint-solving approaches. In the following, we will focus on specific properties of these implicant maps.

For each component in this map, the comparison between its associated implicant and the initial state then defines sets of *required* and *blocker* components for the use of this implicant. We then define two levels of *consistency* of an implicant map formalizing the absence of conflict between required and blockers components. We then associate these consistency properties to either permissive or asynchronous trajectories.

Definition 2 Given a state x and a set of components $J \subseteq V$, the map $\mathcal{I} : J \to \{0, 1, \bigstar\}^V$ is an *implicant map* of J for the state x if for each component $i \in J$ and each state $y \in \mathcal{I}(i)$ we have $f_i(y) \neq x_i$.

Given the state x and a subspace t, we consider the three following sets of components, which form a partition of V:

$$\Delta(x, t) = \{j \in V \mid t_j = \neg x_j\},$$
$$\nabla(x, t) = \{j \in V \mid t_j = x_j\},$$
$$\Psi(t) = \{j \in V \mid t_j = \bigstar\}.$$

Observe that the state x is in the subspace t if and only if $\Delta(x, t) = \emptyset$.

Given an implicant map $\mathcal{I} : J \to \{0, 1, \bigstar\}^V$ for a state x and a set of components $J \subseteq V$, we call $\Delta(x, \mathcal{I}(i))$ the set of *direct requirements* of the component $i \in J$ associated to $\mathcal{I}$, and $\nabla(x, \mathcal{I}(i)) \setminus \{i\}$ its set of *blockers*.

The set of *strong requirements* $\Delta^+(x, \mathcal{I}(i))$ of the component i combines the set of requirements of i with the set of components blocked by i: $\Delta^+(x, \mathcal{I}(i)) = \Delta(x, \mathcal{I}(i)) \cup \{j \neq i \mid i \in \nabla(x, \mathcal{I}(j))\}$.

Intuitively, we want to establish if an implicant map defines a geodesic from x to $\bar{x}^J$. $\Delta(x, \mathcal{I}(i))$ is the set of components that need to change to enable a change in component i. On the other hand, some components can only be updated before a change in component i, thus creating some potential "conflicts" that forbid some updating orders. The sets Δ^+ capture these possible conflicts. To talk about absence of conflicts we introduce the notion of *consistency*.

We need two additional auxiliary constructions. Given an implicant map $\mathcal{I} : J \to \{0, 1, \bigstar\}^V$, define the graphs $G(\mathcal{I}, x)$ and $G^+(\mathcal{I}, x)$ with vertex set J and edge set $\{(j, i) \mid j \in \Delta(x, \mathcal{I}(i))\}$ and $\{(j, i) \mid j \in \Delta^+(x, \mathcal{I}(i))\}$ respectively.

For all $i \in J$, define the sets

$$\overrightarrow{\Delta}(\mathcal{I}, x, i) = \{j \in J \mid \text{there is a path of length greater than zero from } j \text{ to } i \text{ in } G(\mathcal{I}, x)\},$$
$$\overrightarrow{\Delta^+}(\mathcal{I}, x, i) = \{j \in J \mid \text{there is a path of length greater than zero from } j \text{ to } i \text{ in } G^+(\mathcal{I}, x)\}.$$

We call $\overrightarrow{\Delta}(\mathcal{I}, x, i)$ the set of *full requirements* of i and $\overrightarrow{\Delta^+}(\mathcal{I}, x, i)$ the set of *strong full requirements* of i.

An implicant map $\mathcal{I}$ is *consistent* if for each $i \in J$ we have $\overrightarrow{\Delta}(\mathcal{I}, x, i) \subseteq J \setminus \{i\}$.

An implicant map $\mathcal{I}$ is *strongly consistent* if for each $i \in J$ we have $\overrightarrow{\Delta^+}(\mathcal{I}, x, i) \subseteq J \setminus \{i\}$.

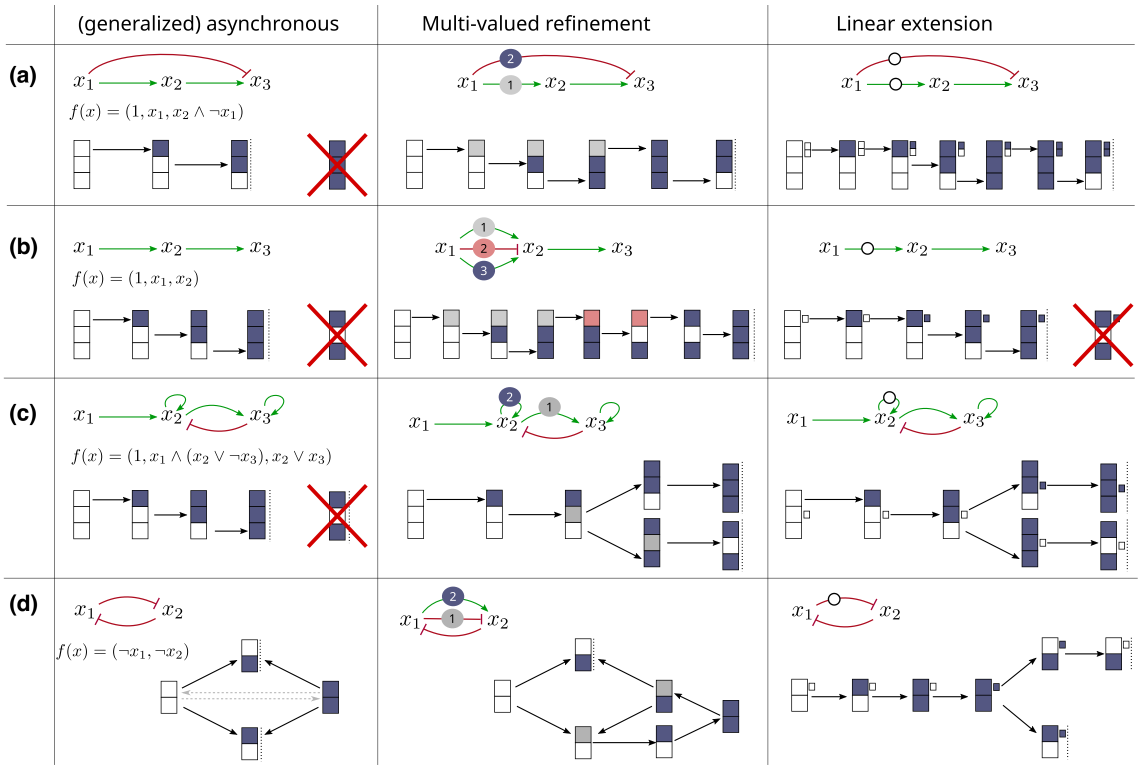

Fig. 1 Reachability properties in Boolean, refined and extended networks. Each row shows a Boolean network with its asynchronous dynamics (left), one of its multi-valued refinements (center) and linear extensions (right). White circles in the extended network denote intermediate linear variables, whereas numbered coloured circles in the refined network denote regulatory thresholds. Selected dynamical trajectories are depicted below each interaction graph. Groups of color-coded squares represent the states of all variables: white for level 0, blue for level 1 (or max), gray and red denote intermediate levels in refinements. Fixed points are marked with a dotted line on the right. The values of intermediate linear variables are represented with smaller squares on the right side of their regulators. **a** An inconsistent feedforward loop: the first component has opposite (direct and indirect) effects on the last one. This competition can be relaxed by associating a higher threshold (center) or adding an intermediate component (left) to the direct interaction. **b** A chain propagating an activation. In the most permissive semantics and some non-monotonic refinements (center), intermediate components can be disabled after propagating the signal. This behaviour can often be considered as an artefact and can not be reproduced in linear extensions. **c** A chain with stabilizing feedback loops. This is an extension of the previous example where feedback loops are added to stabilize the unexpected (1, 0, 1) state. This state is still unreachable in the Boolean network, however it can now be reached in monotonic (single threshold) refinements and in linear extensions. **d** A positive circuit showing that the reachability of the generalized asynchronous (where transitions can involve multiple components) can be reproduced in linear extensions, however it may not be faithfully reproduced in multi-valued refinements

The following result establishes that the conditions of consistency and strong consistency exactly characterize the ability of an implicant map to define a permissive geodesic or an asynchronous geodesic.

Proposition 3 *Given a state x and a set of components $J \subseteq V$, there is a permissive geodesic from x to $\bar{x}^J$ if and only if there if a consistent implicant map of J for x.*

Furthermore, there is an asynchronous geodesic from x to $\bar{x}^J$ if and only if there is a strongly consistent implicant map of J for x.

Proof Take a geodesic $x = x^0, x^1, \ldots, x^l = \bar{x}^J$ in the asynchronous dynamics with direction sequence $i_0, \ldots, i_{l-1}$. Consider the map $\mathcal{I} : J \to \mathbb{B}^V$ defined by $\mathcal{I}(i_k) = x^k$ for $k = 0, \ldots, l-1$ (i.e. the map that associates each component involved in the geodesic with the state in which it changes). Observe that this map is an implicant map of J for x. For each $k = 0, \ldots, l-1$ we have

$$\Delta(x, \mathcal{I}(i_k)) = \{i_0, \ldots, i_{k-1}\},\ \nabla(x, \mathcal{I}(i_k)) = J \backslash \{i_0, \ldots, i_{k-1}\},$$
$$\Delta^+(x, \mathcal{I}(i_k)) = \Delta(x, \mathcal{I}(i_k)) \cup \{j \neq i_k \mid i_k \in \nabla(x, \mathcal{I}(j))\} = \{i_0, \ldots, i_{k-1}\}.$$

It follows that $\overrightarrow{\Delta}(\mathcal{I}, x, i_k) = \overrightarrow{\Delta^+}(\mathcal{I}, x, i_k) = \{i_0, \ldots, i_{k-1}\}$. The map $\mathcal{I}$ is thus strongly consistent.

Now we take a permissive geodesic $x = x^0, x^1, \ldots, x^l = \bar{x}^J$. By definition, for all $k = 0, \ldots, l-1$, there exists $y^k \in [\{x^0, \ldots, x^k\}]$ such that $f_{i_k}(y^k) \neq y^k_{i_k}$. Take the map $\mathcal{I} : J \to \mathbb{B}^V$ defined by $\mathcal{I}(i_k) = y^k$ for all $k = 0, \ldots, l-1$. Observe that $\mathcal{I}$ is an implicant map of J for x. In addition, for all $k = 0, \ldots, l-1$, since $\Delta(x, \mathcal{I}(i_k)) \subseteq \{i_0, \ldots, i_{k-1}\}$, we have $\overrightarrow{\Delta}(\mathcal{I}, x, i_k) \subseteq \{i_0, \ldots, i_{k-1}\}$. Hence the map $\mathcal{I}$ is consistent.

Consider $\mathcal{I} : J \to \{0, 1, \bigstar\}^V$ a strongly consistent implicant map of J for x and $G^+(\mathcal{I}, x)$ the associated graph. Since $\mathcal{I}$ is strongly consistent, we have $i \notin \overrightarrow{\Delta^+}(\mathcal{I}, x, i)$ for all $i \in J$, that is, $G^+(\mathcal{I}, x)$ has no cycle. Hence $G^+(\mathcal{I}, x)$ admits a topological ordering $i_1, \ldots, i_l$. By definition, for each $k \in \{1, \ldots, l\}$ the sub-ordering $i_1, \ldots, i_{k-1}$ contains all components in $\overrightarrow{\Delta^+}(\mathcal{I}, x, i_k)$. For all $h = 1, \ldots, k-1$, since i_h precedes i_k in the ordering, we have that $i_k \notin \Delta^+(x, \mathcal{I}(i_h))$. In particular, i_h is not in $\nabla(x, \mathcal{I}(i_k))$, and is in either $\Delta(x, \mathcal{I}(i_k))$ or $\Psi(\mathcal{I}(i_k))$. Then for each $k = 1, \ldots, l$ we have $\bar{x}^{\{i_1, \ldots, i_{k-1}\}} \in \mathcal{I}(i_k)$, thus the ordering defines the asynchronous geodesic $x, \bar{x}^{\{i_1\}}, \bar{x}^{\{i_1, i_2\}}, \ldots, \bar{x}^J$.

The proof for the permissive geodesic case proceeds similarly, with the sets of full requirements replacing the sets of strong full requirements and $G(\mathcal{I}, x)$ replacing $G^+(\mathcal{I}, x)$. □

Example 4 Consider the Boolean network of Example 2. Take $x = 10$. Define an implicant map of $J = \{1, 2\}$ at x as follows: $\mathcal{I}(1) = 10$, $\mathcal{I}(2) = 10$. Then the graph $G(\mathcal{I}, x)$, with set of vertices $\{1, 2\}$, has no edges, whereas the graph $G^+(\mathcal{I}, x)$ contains the edges $1 \to 2$ and $2 \to 1$. Therefore the implicant map $\mathcal{I}$ is consistent but not strongly consistent. It identifies two permissive geodesics, 10, 00, 01 and 10, 11, 01, corresponding to the two topological orders of $G(\mathcal{I}, x)$.

Let $\mathcal{I}$ and $\mathcal{I}'$ be two different implicant maps for J in x. We say that $\mathcal{I}'$ is a *generalization* of $\mathcal{I}$ if for each $i \in J$ we have $\mathcal{I}(i) \subseteq \mathcal{I}'(i)$. Observe that if $\mathcal{I}$ is (strongly) consistent, then all its generalizations are also (strongly) consistent. We say that $\mathcal{I}$ is a *prime* implicant map if it has no generalization. Observe that if $\mathcal{I}$ is a prime implicant map, then for each $i \in J$, $\mathcal{I}(i)$ is a prime implicant of the function f_i or of its negation (depending on the value of x_i). In this case, the sets of requirements and blockers, and by extension the (strong) full requirements, associated to each component are minimal.

Lemma 4 *If $\mathcal{I}$ is a prime implicant map of J for x, given $i \in J$:*

(i) *for all $j \in \Delta(x, \mathcal{I}(i))$, the interaction graph of f has an edge from j to i;*
(ii) *if $i \in \nabla(x, \mathcal{I}(j))$ for some $j \in J$, then the interaction graph of f has an edge from i to j. In particular, for all $j \in \Delta^+(x, \mathcal{I}(i)) \setminus \Delta(x, \mathcal{I}(i))$ the interaction graph of f has an edge from i to j.*

Proof

(i) By definition of implicant map, for all $y \in \mathcal{I}(i)$ we have $f_i(y) \neq x_i$. Consider $j \in \Delta(x, \mathcal{I}(i))$ and suppose that $j \notin R(i)$. Then $\mathcal{I}_j(i) \neq \bigstar$ and $f_i(\bar{y}^j) \neq x_i$ for all $y \in I(i)$. Then the implicant map $\mathcal{I}'$ defined by $\mathcal{I}'_j(i) = \bigstar$, $\mathcal{I}'_k(i) = \mathcal{I}_k(i)$ for all $k \neq j$ and $\mathcal{I}'(h) = \mathcal{I}(h)$ for all $h \neq i$ is a generalization of $\mathcal{I}$, which contradicts the hypothesis.
(ii) If j is such that $i \in \nabla(x, \mathcal{I}(j))$, then $\mathcal{I}_i(j) \neq \bigstar$ and $f_j(y) \neq x_j$ for all $y \in \mathcal{I}(j)$. If i is not a regulator of j, then $f_j(\bar{y}^i) \neq x_j$ for all $y \in \mathcal{I}(j)$ and $\mathcal{I}$ admits a generalization as in the previous point. □

The following proposition is a corollary of the lemma. Here, given a directed graph G, we write $\tilde{G}$ for the undirected graph obtained by ignoring the directions of all edges.

Proposition 5 *Consider a Boolean network (V, f) with interaction graph G and $\mathcal{I} : J \to \{0, 1, \bigstar\}^V$ a prime implicant map for x. Then, for all $i, j \in V$:*

(i) *if $j \in \overrightarrow{\Delta}(\mathcal{I}, x, i)$ then there is a path of length greater than zero from j to i in G;*
(ii) *if $j \in \overrightarrow{\Delta^+}(\mathcal{I}, x, i)$ then there is a path of length greater than zero from j in i in $\tilde{G}$; if $j \in \overrightarrow{\Delta^+}(\mathcal{I}, x, i) \setminus \overrightarrow{\Delta}(\mathcal{I}, x, i)$ then there is at least one edge (h, k) in the path such that (k, h) is an edge in G.*

Proof

(i) By Lemma 4(i), $G(\mathcal{I}, x)$ is a subgraph of G. The conclusion follows from the definition of $\overrightarrow{\Delta}(\mathcal{I}, x, i)$.
(ii) By Lemma 4(i) and (ii), $G^+\widetilde{(\mathcal{I}, x)}$ is a subgraph of $\tilde{G}$, hence the first part of the statement. If j is in $\overrightarrow{\Delta^+}(\mathcal{I}, x, i)$ but not in $\overrightarrow{\Delta}(\mathcal{I}, x, i)$, then at least one of the edges (h, k) in the path satisfies $h \in \Delta^+(x, \mathcal{I}(k)) \setminus \Delta(x, \mathcal{I}(k))$, and we conclude using Lemma 4(ii). □

4 *L*-cuttable Boolean networks

In the previous section, we identified conditions on the implicant maps associated to a given initial state for the existence of a geodesic in permissive trajectories or in classical asynchronous trajectories. In presence of a permissive geodesic, we observed that conflicts captured by the implicant map and the associated auxiliary graph can prevent the existence of the corresponding asynchronous geodesic. Here we will define a topological class of networks in which such conflicts do not exist. In this case, all consistent implicant maps are also strongly consistent, and thus all permissive geodesics exist in the asynchronous dynamics.

In the following, we say that a component of a network is *linear* if it has a single regulator and a single target. In the next definition we introduce the class of *linearly-cuttable* networks, that is, networks that admit a set of linear components separating all potential regulatory conflicts. We will show that, for asynchronous dynamics associated to linearly-cuttable Boolean networks, trap spaces provide good approximation of attractors; in addition, we will prove some general reachability properties.

Definition 3 Given a directed graph G on V, a *linear cut* of G is a set $L \subseteq V$ of linear components such that

(i) every cycle in G contains at least one component of L,

(ii) every path of length greater than zero in G from a component with multiple targets to a component with multiple regulators contains at least a component of L.

A linear cut L is *minimal* if there is no linear cut strictly included in L.

A Boolean network $M = (V, f)$ is *L-cuttable* if $L \subseteq V$ is a linear cut for its interaction graph G.

If a Boolean network is L-cuttable for some $L \subseteq V$, we say that the network is linearly-cuttable or simply cuttable for brevity. We will also need the notion of *canonical states*. For an L-cuttable network (V, f), a state $x \in \mathbb{B}^V$ is *L-canonical* if for each $i \in L$ we have $f_i(x) = x_i$. That is, a state x is L-canonical if all components in L are stable in x.

Example 5 The Boolean network $f(x_1, x_2) = (x_2, x_1)$ is cuttable. Taking for instance $L = \{2\}$, the state 00 is L-canonical, and the state 01 is not L-canonical.

Example 6 The Boolean network of Example 3 and Fig. 1a is not L-cuttable for any subset L of $V = \{1, 2, 3\}$: the path $1 \rightarrow 3$ in the interaction graph of f from variable 1 (having two targets) to variable 3 (having two regulators) does not contain any linear variable, hence condition (*ii*) of Definition 3 cannot be satisfied.

Remark 1 If G admits a loop, that is, a cycle of length one $i \rightarrow i$, then by point (i) of Definition 3, i belongs to L. Then i admits only one regulator and one target, which means that $i \rightarrow i$ is an *isolated loop*.

Suppose that g is Boolean network with interaction graph H and f is obtained from g by adding the self-regulated component i.

(1) If the loop has a negative sign, then the asynchronous dynamics of f contains two copies of the asynchronous dynamics of g, one contained in the subspace $\{x_i = 0\}$ and one in the subspace $\{x_i = 1\}$. The two copies are connected by transitions along variable i in both directions: for each state x, there are transitions $x \rightarrow \bar{x}^i$ and $\bar{x}^i \rightarrow x$. Note that, in this case, no state is L-canonical.

(2) If the loop has a positive sign, then the asynchronous dynamics of f consists of the disjoint union of two graphs that are copies of the asynchronous dynamics of g, one contained in the trap space $\{x_i = 0\}$, one in the trap space $\{x_i = 1\}$.

These observations allows us to focus the study of cuttable networks to graphs that do not have any loop, since the properties are easily transferable to the general case.

Remark 2 Consider a linear cut L and two distinct vertices i, j in L. Suppose that there is an edge from i to j. Since i is the unique regulator of j, any cycle containing j must also contain i, and any path from a component with multiple targets to a component with multiple regulators containing j must also contain i. As a consequence, $L \backslash \{j\}$ is also a linear cut for G.

Suppose now that the linear cut L is minimal with respect to inclusion. If G has no isolated loop, the previous observation shows that L is an independent set of G, that is, no pair of vertices in L is connected by an edge.

We now prove properties of implicant maps for networks with linear cuts.

Remark 3 Consider x L-canonical and $\mathcal{I} : J \rightarrow \{0, 1, \star\}^V$ a prime implicant map for x and $i \in J \cap L$. Since i has only one regulator j, if $j \in J$ we must have $\mathcal{I}_j(i) = \bar{x}_j^j$ and $\mathcal{I}_k(i) = \star$ for all $k \neq j$, which gives $\Delta(x, \mathcal{I}(i)) = \{j\}$, $\Psi(\mathcal{I}(i)) = V \backslash \{j\}$ and $\nabla(x, \mathcal{I}(i)) = \emptyset$.

Lemma 6 *Given an L-canonical initial state x in an L-cuttable network, all prime generalizations of consistent implicant maps for x are strongly consistent.*

Proof Consider a consistent implicant map $\mathcal{I}'$ and take a generalisation $\mathcal{I}$ of $\mathcal{I}'$ that is prime. Suppose that $\mathcal{I}$ is a consistent but not strongly consistent implicant map for x, i.e., there is at least one component i such that $i \in \overrightarrow{\Delta^+}(\mathcal{I}, x, i) \backslash \overrightarrow{\Delta}(\mathcal{I}, x, i)$. By Proposition 5 (*ii*), i is part of a cycle in $\tilde{G}$, with at least one edge (j, k) such that $j \in \Delta^+(x, \mathcal{I}(k)) \backslash \Delta(x, \mathcal{I}(k))$ and $(k, j) \in G$ (at least one edge is associated to a blocker).

If all edges are associated to blockers, the cycle is also a cycle in G. By definition of L-cuttable network, this cycle contains at least one component of L. As x is canonical, the components of L have no blockers (Remark 3) and we have a contradiction.

Thus the cycle contains at least one edge associated to a direct requirement and another edge (j, k) associated to a blocker. Take the maximal sub-path π in the cycle that contains (j, k) and is composed of edges associated to blockers, and call j' and k' the first and last vertex in the path. Then G contains edges (j'', j') and (k', k'') that are not part of π, and since the path π is associated to blockers, G contains a path from k' to j'. That is, the reverse π' of the path π is a path in G from a vertex with multiple targets (k') to a vertex with multiple regulators (j'). By definition of L-cuttable network, π' contains an element of L. Since all edges of π' are associated to blockers, this is again in contradiction with Remark 3. □

By combining the lemma with Propositions 2 and 3, we derive some corollaries.

Corollary 7 *Let (V, f) be a Boolean network that admits a linear cut $L \subseteq V$ and let x be an L-canonical state. If there exists a permissive geodesic from x to the state y, then there exists also an asynchronous geodesic from x to y. In particular*:

(i) *$[x, y]$ is the minimal trap space containing x if and only if there exists a maximal geodesic from x to y.*
(ii) *for all subsets of components $J \subseteq \Delta(x, f(x))$ there exists a path from x to $\bar{x}^J$ (in other words, all the successors of x in the generalized asynchronous dynamics are reachable from x in the asynchronous dynamics).*
(iii) *The smallest subspace containing the states that are reachable from x is a trap space.*
(iv) *If y is the last vertex of a geodesic starting from x and $f_i(z) \neq z_i$ for some $z \in [x, y]$* and *$i \notin \Delta(x, y)$, then there is a geodesic from x to $\bar{y}^i$.*

The corollary gives information on the set of states that are reachable from canonical states in asynchronous dynamics. It tells us that this set is as large as it can be, as allowed by the most permissive semantics. For example, the image $f(x)$ of a canonical state x is reachable from x, and the farthest state in the minimal trap space containing x is reachable from x.

Remark 4 In Noual and Sené (2018), a weaker sufficient condition for (ii) is given. Let (V, f) be a Boolean network, and let D be its signed interaction graph: the vertex set is V and, for all $i, j \in V$, there is a positive (negative) edge from i to j if there exists $x \in \mathbb{B}^V$ with $x_i = 0$ such that $f_j(\bar{x}^i) - f_j(x)$ is positive (negative); the presence of both a positive and a negative edge from i to j is allowed. For all $x \in \mathbb{B}^V$, let $D(x)$ be the spanning subgraph of D that contains all the positive (negative) edges (i, j) of D such that $x_i \neq x_j$ ($x_i = x_j$). Noual and Sené (2018) (Proposition 2) proved that (ii) holds whenever the subgraph of $D(x)$ induced by $\Delta(x, f(x))$ has no cycle. Suppose now that x is L-canonical, and let us prove that this forces $D(x)$ to be acyclic (and so the condition of Noual and Sené is indeed weaker than ours). Let $i \in L$ and j its unique in-neighbor. Then either $f_i(y) = y_j$ for all $y \in \mathbb{B}^V$ or $f_i(y) = \neg y_j$ for all $y \in \mathbb{B}^V$. In the first (second) case (j, i) is positive (negative) and $x_i = f_i(x) = y_j$ ($x_i = f_i(x) \neq y_j$) since x is L-canonical. So in both cases, (j, i) is not an edge of $D(x)$, and thus i has in-degree zero in $D(x)$. Since each cycle of D has a vertex in L, we deduce that $D(x)$ is acyclic.

Example 7 Consider again the Boolean network of Examples 2 and 4. The Boolean network is L-cuttable with $L = \emptyset$, and the state $x = 10$ is L-canonical. The implicant map $\mathcal{I}$ defined in Example 4 admits the prime generalisation $\mathcal{I}'$ with $\mathcal{I}'(1) = \star\star$, $\mathcal{I}'(2) = 1\star$. Since $\mathcal{I}$ is consistent, by Lemma 6 $\mathcal{I}'$ is strongly consistent. In fact, $\Delta(x, \mathcal{I}'(1)) = \Delta(x, \mathcal{I}'(2)) = \nabla(x, \mathcal{I}'(1)) = \emptyset$, $\nabla(x, \mathcal{I}'(2)) = \{1\}$, and the graph $G^+(x, \mathcal{I}')$ admits only one edge, $2 \to 1$. Therefore to construct an asynchronous geodesic from 10 to 01 one must first update the second variable, then the first, obtaining the sequence 10, 11, 01.

Example 8 Consider the Boolean network with three components $g(x_1, x_2, x_3) = (x_2, x_3, x_1)$. It is L-cuttable, with, for instance, $L = \{3\}$. We can view g as an extended version of the Boolean network f in Example 5, obtained by adding variable 3 as an intermediate linear variable between variables 1 and 2 (we discuss linear extensions of Boolean networks in detail in Sect. 5). For this linear cut, the state 010 is canonical. The corollary tells us for instance that the state $g(010) = 100$ is reachable from 010 in the asynchronous dynamics of g, and that so is the canonical state 101. Note that the state 10 is not reachable from the state 01 in the asynchronous dynamics of f (while it is reachable in the generalized asynchronous and in the most permissive semantics). We can therefore say that the addition of variable 3, while not changing the interaction graph substantially, has enriched the reachability properties of the asynchronous dynamics.

The conclusions of Corollary 7 do not hold for states that are not canonical: for instance, in the asynchronous dynamics of the $\{2\}$-cuttable Boolean network of Example 5 ($f(x_1, x_2) = (x_2, x_1)$) there are no paths from the non-canonical state 01 to 10, while there are transitions to 00 and 11, so that $[01] = \mathbb{B}^2$. The same example also shows that point (*i*) of Definition 3 cannot be relaxed.

Corollary 8 *Let (V, f) be a Boolean network that admits a linear cut $L \subseteq V$.*

(i) *The smallest subspace containing an attractor is a trap space.*

(ii) *If x is an L -canonical state that belongs to an attractor A, there is a geodesic from x to* $\bar{x}^{\Delta(A)}$, *and* $\bar{x}^{\Delta(A)}$ *is L -canonical.*

The corollary describes a property that all attractors of linearly-cuttable networks must satisfy: all free variables of the minimal trap space containing an attractor must oscillate in the attractor. This feature becomes particularly useful when combined with a property that we will prove in Sect. 4.2, which will give uniqueness of attractors in minimal trap spaces.

Example 9 Consider the Boolean network defined by

$$f(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (\neg x_6, x_1, \neg x_2, x_3, \neg x_4, x_5, x_1 \wedge x_3 \wedge x_5).$$

It is easy to see that the asynchronous dynamics of f has a cyclic attractor A containing the state 1100000, and that oscillates in all variables except for variable 7. That is, the set $\{x \in B^7 | x_7 = 0\}$ is the minimal subspace $[A]$ containing A. Since the state 1111111 is a successor of 1111110 in the asynchronous dynamics, $[A]$ is not a trap space: the minimal trap space containing A is $\mathbb{B}^7$. The Boolean network is in fact not linearly-cuttable: there are edges from 1, 3 and 5 (variables with multiple targets) to 7 (variable with multiple regulators). Corollary 8 suggests a way to modify the Boolean network to obtain an attractor that oscillates in all variables. It states that this can be achieved by adding linear variables between 1 and 7, 3 and 7 and 5 and 7, obtaining a network of the form

$$g(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}) = (\neg x_6, x_1, \neg x_2, x_3, \neg x_4, x_5, x_8 \wedge x_9 \wedge x_{10}, x_1, x_3, x_5).$$

4.1 Reachability of trap spaces from canonical states

In this section we continue the analysis of reachability properties from canonical states in the asynchronous dynamics. We recall that given an initial state x, the minimal trap space t containing x contains all states reachable from x for any updating (i.e. it gives an upper bound on the distance of reachable states). It is in general very difficult to deduce further information on the states reachable from x from the structure of the interaction graph. In this section we show that the existence of a linear cut guarantees the reachability of all trap spaces contained in t if x is a canonical state (Fig. 2).

The full significance of this statement becomes apparent when we combine it with the result of the next section: we will see that, in networks with linear cuts, all attractors are contained in minimal trap spaces. For a canonical initial condition this means that all the attractors contained in the minimal trap space containing the initial condition are reachable. We have therefore a connection between a structural property of the interaction graph and a reachability property that is far from trivial.

For the proof of the main result of this section we need the following straightforward lemma.

Lemma 9 *Let* (V, f) *be a Boolean network and P a geodesic from x to y with direction sequence w. Let* $i \in \Delta(x, y)$ *and suppose that G has no edge from i to a vertex that appears after i in w. Then there exists a geodesic from x to* $\bar{y}^i$ *whose direction sequence is obtained from w by deleting i.*

The idea of the proof of the following theoreom is that, starting from a canonical state x, we can first reach the opposite state in the minimal trap space containing x, and then, using the flexibility given by the intermediate linear variables, we can undo all changes that took the trajectory away from a given target trap space.

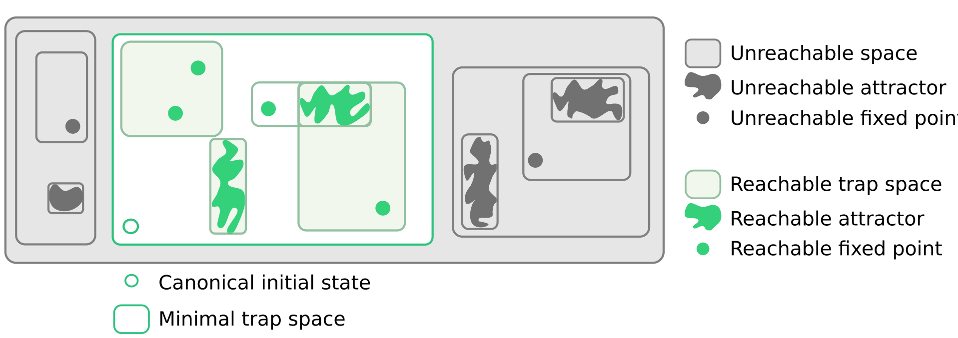

Fig. 2 Summary of the reachability of trap spaces and attractors. Given an initial state, all states, and in particular all attractors, that are not contained in the minimal trap space containing the initial state are not reachable in any updating semantic. For L-cuttable networks and L-canonical initial states, all trap spaces and attractors included in the minimal trap space are reachable

Theorem 10 *Let (V, f) be a Boolean network with interaction graph G and $L \subseteq V$ a linear cut. Let x be an L-canonical state and $[x, y]$ be the minimal trap space containing x. For every trap space $t \subseteq [x, y]$ there is a path in the asynchronous dynamics from x to a state in t of length at most $2n$.*

Proof First observe that, if G has a loop, by Remark 1 it must be an isolated loop. If the loop is negative, by Remark 1(1) there is nothing to show. If the loop is positive, then it is sufficient to prove the theorem for the Boolean network obtained by removing the isolated loop. Hence we can assume that G has no loops.

Define $J = \Delta(x, t) \subseteq \Delta(x, y)$. By Corollary 7 (i), since x is canonical, there is a geodesic from x to y. Take $z \in [x, y]$ such that $J \subseteq \Delta(x, z)$, there is a geodesic P from x to z and the cardinality of $K = \Delta(z, t)$ is minimal. This set is non-empty since it contains y. We will show that there is a path from z to $\bar{z}^K$, which belongs to t.

Since $K \cap J = \emptyset$, for any $i \in K$ we have $z_i \neq x_i$ and thus i appears in the direction sequence of P. Let $i_0, \ldots, i_{l-1}$ be an enumeration of K as in the direction sequence of P.

We first prove the following property.

(I) *There is no $0 \leq p \leq q \leq l$ such that G has an edge from i_q to i_p.*

Suppose, for a contradiction, that there is $1 \leq p \leq q \leq l$ such that G has an edge from i_q to i_p. Since G has no loop we have $p < q$. Suppose first that i_p has only one regulator. Since i_q is in K and not in J, we have $x_{i_q} = t_{i_q}$ and since t a trap space and i_q is the unique regulator of i_p, we derive $f_{i_p}(x) = x_{i_p}$. Then i_q appears before i_p in the direction sequence of P, a contradiction. So i_p has at least two regulators. Since G has a linear cut, i_p is the unique target of i_q, and we deduce from Lemma 9 that there is a geodesic from x to $\bar{z}^{i_q}$. Since i_q is in K, this contradicts the minimality of K.

Let us prove that there is a geodesic $z = z^0, z^1, \ldots, z^l = \bar{z}^K$ from z to $\bar{z}^K$ with direction sequence $i_0, \ldots, i_{l-1}$. We have to prove that $f_{i_k}(z^k) \neq z^k_{i_k}$ for $0 \leq k < l$. Since t is a trap space, $f_j(w) = w_j \neq z_j$ for all $j \in K$ and $w \in t$, therefore it is sufficient to show that $z^k_j = t_j$ for any regulator j of i_k.

We proceed by induction on k. Let j be a regulator of i_0. By (I) we have $j \notin K$, so $z^0_j = z_j = t_j$. Let $0 < k < l$ and let j be a regulator of i_k. If $j \notin K$, then $z^k_j = z_j = t_j$ by definition of K. Otherwise, by (I) we have $j \in \{i_0, \ldots, i_{k-1}\}$ thus $z_j \neq z^k_j$, and we deduce that $z^k_j = t_j$. □

Example 10 Consider the BN with 8 components defined by $f(x) = (x_8, x_8, x_7, x_5, x_6 \wedge \neg x_1, x_7, x_3 \vee \neg x_2, \neg x_4)$. Take the state x with $x_i = 0$ for $i = 1, \ldots, 8$. One can easily see that the minimal trap space containing x is $\mathbb{B}^8$, either by observing that, by choosing the right order, all components can be updated to 1, or using for example one of the techniques of Klarner et al. (2015), Chevalier et al. (2019) and Trinh et al. (2022). Observe that f is L-cuttable with $L = \{1, 2, 3, 4, 6\}$, and that x is L-canonical. Theorem 10 then gives that all fixed points are reachable from x.

Example 11 The theorem does not hold if the initial state is not L-canonical. The Boolean network $f(x_1, x_2, x_3, x_4, x_5) = (x_3, x_4 \wedge x_5, x_1, x_1, x_2)$ is L-cuttable with $L = \{3, 4, 5\}$. The fixed points of f are 00000, 10110 and 11111.

Consider the state $x = 11011$, which is not L-canonical ($(f(x)_3 \neq x_3$). The fixed point 11111 is a direct successor for x in the asynchronous dynamics of f. In addition, 00000 is reachable from x in the asynchronous dynamics of f via the path $11011 \to 01011 \to 01001 \to 00001 \to 00000$. As a consequence, the minimal trap space containing x is the full space $\mathbb{B}^5$. Observe that there is no path from 11011 to the fixed point 10110.

4.2 Minimal trap spaces are good approximations for attractors

In this section we prove that, in asynchronous dynamics of linearly-cuttable networks, attractors and minimal trap spaces are in one-to-one correspondence, thus adding to the catalogue of structural properties that determine features of the dynamics. When attractors are confined in minimal trap spaces, not only is their identification simplified, but also properties like controllability are easier to study. Although Boolean models might often not come equipped with a linear cut, it is easy to extend a model to satisfy the linear cut property (see Sect. 5) while preserving the structure of cycles and their signs, meaning that linear cuts can be useful tools to keep in mind when designing models.

Moreover, known results that can be used to draw conclusions about the dynamics from the interaction structure of a Boolean network usually require strong conditions on the cycles, and mainly concern the existence and number of attractors (see Richard 2018 for a review of the main results in this direction). Thanks to the generality of the assumptions in terms of cyclic structure and asymptotic dynamics (no restriction is posed on network motifs or the nature of the attractors), from a theoretical point of view the results we discuss bring new perspectives to the study of the relationship between structure and dynamics.

Theorem 11 *Suppose that (V, f) is L-cuttable and A is an attractor for the asynchronous dynamics of f. Then $[A]$ is a trap space and, for every $x \in [A]$, there is a geodesic from x to A.*

We prove not only that an attractor A of a cuttable network can be reached from any state in $[A]$, but also that it can be reached with the smallest number of steps possible. We saw in Corollary 8 that, in Boolean networks with linear cuts, the minimal subspace containing an attractor is a trap space, and that, from any canonical state in the attractor, the state at the opposite end of the enclosing trap space can be reached via a geodesics. It is by exploiting the existence of such geodesics, showing that we can "copy" the right steps from them, that we build a geodesic from a generic state in the enclosing trap space to the attractor.

Given two states x, y, we set $[x,y[=[x,y]\backslash\{y\}$.

Lemma 12 *Suppose that (V,f) is L -cuttable with L minimal cut, and that the interaction graph of f has no loops. Let A be an attractor for the asynchronous dynamics of f. Consider $x\in[A]$ and $y\in A$, and suppose that y is L -canonical. Let I be the set of $i\in L$ with $f_i(x)\neq x_i=y_i$. Suppose that there is no L -canonical state in $[\bar{x}^I,y[\cap A$. Then there is a geodesic from x to y.*

Proof We proceed by induction on the Hamming distance $d(x,y)=|\Delta(x,y)|$. If $d(x,y)=0$ there is nothing to prove, so suppose that $d(x,y)>0$. As a first step in the proof, we show that there is no transition from y towards x involving a variable that is not in L (point (1)), as this would give a canonical state in the attractor that is closer to x than y. Second, using point (1), we prove that from x we can take a step towards y, using the existence of geodesics in the attractor. From this, we show that we can apply the induction hypothesis.

(1) *There is no $i\in\Delta(x,y)\backslash L$ such that $\bar{y}^i\in A$.*

Suppose, for a contradiction, that $z=\bar{y}^i$ is in A for some $i\in\Delta(x,y)\backslash L$. Let J be the targets j of i such that $j\in L$ and $f_j(z)\neq z_j$. Since y is L-canonical and L, by Remark 2, is an independent set, there is a geodesic from z to $\bar{z}^J$, which is L-canonical. Suppose, for a contradiction, that $\bar{z}^J\notin[\bar{x}^I,y]$. Then there is a component j such that $\bar{z}^J_j\neq\bar{x}^I_j=y_j$. Since $x_k=y_k$ for all $k\in I$, we have $j\notin I$, thus $\bar{z}^J_j\neq x_j=y_j$. Since $x_i\neq y_i$ we have $j\neq i$, thus $\bar{x}^J_j\neq x_j=y_j$. We deduce that $j\in J$. Since $x_i=z_i\neq y_i$ and since y is L-canonical, we have $f_j(x)=f_j(z)\neq f_j(y)=y_j=x_j$ thus $j\in I$, a contradiction. This proves that $\bar{z}^J\in[\bar{x}^I,y]$, and since $z_i\neq y_i$ we have $z\in[\bar{x}^I,y[$. Since $\bar{z}^J$ is L-canonical and reachable from y, we have $\bar{z}^J\in A$ and we obtain a contradiction.

(2) *$f_i(x)\neq x_i$ for some $i\in\Delta(x,y)$.*

Suppose not, that is, $f_i(x)=x_i$ for all $i\in\Delta(x,y)$. Since y is L-canonical, by Corollary 8 (*ii*), there is a geodesic from y to $y'=\bar{y}^{\Delta(A)}$, and a geodesic P from y' to y. Let i be the first component of the direction sequence of P with $x_i\neq y_i$. Since $x,y\in[A]$, we have $\Delta(x,y)\subseteq\Delta(A)$, thus this component i exists. Let z be the state of P with $f_i(z)\neq z_i$.

Let us prove that i has at least two regulators. Suppose not. Since $i\in\Delta(A)$, i has only one regulator, and its regulator j is in $\Delta(A)$. If $i\in L$ then $f_i(y)=y_i$ since y is L-canonical, and if $i\notin L$, then, since $x_i\neq y_i$, we have $f_i(y)=y_i$ by (1). Since $i,j\in\Delta(A)$, we obtain $f_i(y')=y'_i$. Since $f_i(z)\neq z_i=y'_i$, we have $z_j\neq y'_j$ and thus j appears before i in the direction sequence of P. By the choice of i, we have $x_j=y_j$ and thus $f_i(x)=f_i(y)=y_i\neq x_i$, which contradicts our hypothesis. This proves that i has at least two regulators.

Let P' be the path from $z'=\bar{z}^i$ to y contained in P. Let J be the set of regulators j of i such that $x_j\neq y_j$. We have $i\notin J\subseteq\Delta(x,y)\subseteq\Delta(A)$. Hence, by the choice of i, $J\cap\Delta(y',z')=\emptyset$, and since $\Delta(y',z'),\Delta(z',y)$ is a partition of $\Delta(A)$, we have $J\subseteq\Delta(z',y)$. Hence $\bar{y}^J\in[z',y]$. By the definition of J and our hypothesis, we have $f_i(\bar{y}^J)=f_i(x)=x_i\neq y_i=\bar{y}^J_i$. Since $z'_i=y_i$, we deduce from Corollary 7 (*vi*) that there is a geodesic from z' to $\bar{y}^i$, and since $\bar{y}^i$ is reachable from y, we have $\bar{y}^i\in A$. Since i has at least two regulators, this contradicts (1).

By (2) there is a component i with $x_i\neq f_i(x)=y_i$. Then there is a transition from x to $z=\bar{x}^i$. Let J be the set of $j\in L$ with $f_j(z)\neq z_j=y_j$ (we have $i\notin J$ since otherwise i has a negative loop). Let us prove that $\bar{z}^J\in[\bar{x}^I,y]$.

Take a component j such that $\bar{x}^I_j=y_j$. We have to show that $\bar{z}^J_j=\bar{x}^I_j=y_j$. We have $j\notin I$ by definition of I, hence $x_j=y_j$. Since, by choice of i, $x_i\neq y_i$, we have $j\neq i$, so $z_j=\bar{x}^i_j=x_j=y_j$. Suppose that j is in J, that is, $j\in L$ and $f_j(z)\neq z_j$. Since j is not in I, we have $f_j(x)=x_j$, and i is therefore the unique regulator of j. Since $z_i=y_i$, we have $f_j(y)=f_j(z)\neq y_j$, but then y is not L-canonical, a contradiction. Hence j is not in J and $\bar{z}^J_j=\bar{x}^I_j=y_j$ as wanted.

This proves that $\bar{z}^J\in[\bar{x}^I,y]$ and thus $[\bar{z}^J,y]\subseteq[\bar{x}^I,y]$. Hence, by hypothesis, there is no L-canonical state in $[\bar{z}^J,y[\cap A$. Since $d(z,y)<d(x,y)$, by induction, there is a geodesic from z to y and thus also a geodesic from x to y. □

Lemma 13 *Suppose that (V,f) is L -cuttable with L minimal cut, and that the interaction graph of f has no loops. Let A be an attractor for the asynchronous dynamics of f. Consider $x\in[A]$ and $y\in A$, and suppose that y is L -canonical. Let I be the set of $i\in L$ with $f_i(x)\neq x_i=y_i$. Then there is a geodesic from x to some L -canonical state $a\in[\bar{x}^I,y]\cap A$.*

Proof We proceed by induction on $d(\vec{x}^I, y)$. Since $x_i = y_i$ for all $i \in I$, we have $x \in [\vec{x}^I, y]$, so if $d(\vec{x}^I, y) = 0$ then $x = y$ and there is nothing to prove. So suppose that $d(\vec{x}^I, y) > 0$. If there there is no L-canonical state in $[\vec{x}^I, y[\cap A$, then, by Lemma 12, there is a geodesic from x to y, so the lemma holds with $a = y$. So suppose that there is an L-canonical state $y' \in [\vec{x}^I, y[\cap A$. Let I' be the set of $i \in L$ with $f_i(x) \neq x_i = y'_i$. The idea is to show that $\vec{x}^{I'}$ and y' are strictly closer to each other than $\vec{x}^I$ and y, so that the induction hypothesis can be applied.

We have $[\vec{x}^{I'}, y'] \subseteq [\vec{x}^I, y]$. Indeed, since $y' \in [\vec{x}^I, y]$ it is sufficient to prove that $\vec{x}^{I'} \in [\vec{x}^I, y]$. That is, given i such that $\vec{x}^I_i = y_i$, we have to show that $\vec{x}^{I'}_i = \vec{x}^I_i$. Since, by definition of I, we have $i \notin I$, we just need to show that i is not in I'. Since y' is in $[\vec{x}^I, y]$, we have $y'_i = y_i$; as a consequence, $i \in I'$ would imply $i \in I$, a contradiction.

We have $\Delta(y, y')\backslash L \neq \emptyset$. Indeed, let $i \in \Delta(y, y')$. If $i \notin L$ we are done. So suppose that $i \in L$ and let j be its regulator. Since y', y are L-canonical, $f_i(y') = y'_i \neq y_i = f_i(y)$ thus $y'_j \neq y_j$. Since L is a minimal linear cut, by Remark 2 L is an independent set thus $j \notin L$ and $j \in \Delta(y, y')\backslash L$.

So let $i \in \Delta(y, y')\backslash L$. Since $I', I \subseteq L$ and $y' \in [\vec{x}^I, y]$ we have $y_i \neq y'_i = \vec{x}^I_i = \vec{x}^{I'}_i$, thus $i \in \Delta(\vec{x}^I, y)\backslash\Delta(\vec{x}^{I'}, y')$. Since $[\vec{x}^{I'}, y'] \subseteq [\vec{x}^I, y]$ we have $\Delta(\vec{x}^{I'}, y') \subseteq \Delta(\vec{x}^I, y)$ and we deduce that $d(\vec{x}^{I'}, y') < d(\vec{x}^I, y)$.

Consequently, by induction, there is an L-canonical state $a \in [\vec{x}^{I'}, y'] \cap A \subseteq [\vec{x}^I, y] \cap A$ such that there is a geodesic from x to a. This completes the induction. □

Proof of of Thorem 11 The first part is Corollary 8(i).

For the second part, observe first that, by Remark 1, we can assume that the interaction graph of f has no loops. Taking any minimal cut contained in L, we can conclude by applying Lemma 13, after observing that any attractor of the asynchronous dynamics of f contains at least one canonical state. □

Remark 5 Theorem 11 shows that each minimal trap space in a linearly-cuttable network contains only one attractor in the asynchronous updating. Note that this attractor is not always equal to the enclosing minimal trap space as in the most permissive semantics (Paulevé et al. 2020). For example, a Boolean network with at least three components and interaction graph consisting of a negative cycle is linearly-cuttable yet the minimal trap space contains states that are not part of the enclosed attractor (see Remy et al. 2003, for a full characterisation of the dynamics associated to isolated circuits).

Example 12 Consider again the Boolean network of Example 10. By Theorems 10 and 11, we can conclude that all attractors are reachable from x. For the Boolean network g of Example 9, Theorem 11 gives that the attractor is reachable from all states via a geodesic.

5 Cuttable extended semantics

Given a Boolean network, we obtain an *extended network* by replacing a subset of the interactions with linear components. We show that the trap spaces of the original network are also trap spaces of its extensions, which provide an over-approximation of the original asynchronous dynamics. We will focus on *cuttable extended networks* in which the additional linear components form a linear cut of the extended network. Cuttable extensions allow us to define an execution semantics that takes advantage of the properties of cuttable networks for any Boolean network.

In many applications to biology, Boolean networks are abstract models often used in absence of quantitative knowledge on precise concentrations and kinetic parameters. The non-determinism of the classical asynchronous semantics accounts for this lack of knowledge by enabling alternative trajectories corresponding to quantitative differences in initial conditions and kinetic parameters. However, it assumes that a change of the state of a component is reflected on all its targets at the same time. The introduction of intermediate linear components lets us eliminate this assumption. The alternative trajectories obtained in the asynchronous dynamics of an extended network then cover plausible behaviours that may be missing in the asynchronous dynamics of the original network.

5.1 Definition and properties

Definition 4 Let $M = (V, f)$ be a Boolean network with edges E and $L \subseteq E \subseteq V^2$ a subset of its interactions. Consider the Boolean function $\mathcal{E}(f, L) : \mathbb{B}^{V\cup L} \to \mathbb{B}^{V\cup L}$ defined as follows. For each $i \in V \cup L$

$$\mathcal{E}(f, L)_i(y) = \begin{cases} f_i(\pi^i(y)) & \text{if } i \in V, \\ y_j & \text{if } i = (j,k) \in L, \end{cases}$$

where $\pi^i : \mathbb{B}^{V\cup L} \to \mathbb{B}^V$ is defined for all $j \in V$ as:

$$\pi^i(y)_j = \begin{cases} y_{(j,i)} & \text{if } (j,i) \in L, \\ y_j & \text{otherwise.} \end{cases}$$

We call the Boolean network $(V \cup L, \mathcal{E}(f, L))$ an *extended network* and the *L-extension* of f.

For an extended network $(V \cup L, \mathcal{E}(f, L))$, we call V the set of *core variables* and L the set of *extender variables*. We say that an L-extension is *cuttable* if it is L-cuttable. We call the E-extension of f its *full extension*. By construction, the E-extension is cuttable. We will need the following additional notations. We write $\pi : \mathbb{B}^{V \cup L} \to \mathbb{B}^V$ for the projection onto $\mathbb{B}^V$, and define the map $\epsilon : \mathbb{B}^V \to \mathbb{B}^{V \cup L}$ that "copies" each regulator, once for each of its target variable:

$$\epsilon_k(x) = \begin{cases} x_j & \text{if } k = (j,i) \in L, \\ x_k & \text{otherwise.} \end{cases}$$

Example 13 The Boolean network (V, f) of Examples 3 and 6 and Fig. 1a) can be extended to a cuttable extension by taking $L = \{(1,2),(1,3)\}$. Note that, while in the asynchronous dynamics of f there is no path from 000 to 111, in the asynchronous dynamics of the L-extension the state 11111 is reachable from 00000 via the geodesic that updates the variables in $V \cup L$ the following order: 1, (1, 2), 2, 3, (1, 3).

Note that $x = \pi(\epsilon(x)) = \pi^i(\epsilon(x))$ for any $i \in V$, and that if L contains no interaction with target i, then $\pi^i = \pi$. We call the states $y \in V \cup L$ that satisfy $\epsilon(\pi(y)) = y$ (that is, states for which the extender variables mirror their regulators) *canonical* states. Note that, by construction, all canonical states of an L-extended network are L-canonical.

Aside from the partition of their components into core and extender variables, extended networks are regular networks and the notations introduced above, such as $T(i)$ and $R(i)$, apply as usual. Depending on the context, extender variables will be referred to as regular variables (e.g. $i \in (V \cup L)$) or as a pair of core variables (e.g. $(i,j) \in V^2$).

Definition 5 Let $M = (V,f)$ be a Boolean network, x and y two states of $\mathbb{B}^V$, and L a subset of its interactions. We say that y is *L-reachable* from x if there is a trajectory from $\epsilon(x)$ to $\epsilon(y)$ in the asynchronous dynamics of the L-extension of M.

This definition of L-reachability allows us to study reachability in any Boolean network using canonical initial states in an extended network. Note that the set of states that are reachable from a non-canonical state can differ significantly from the set of states that are reachable from the canonical state that projects to the same core variables. For instance, consider a Boolean network such that all components have at least one regulator, and take the full extension. Then all canonical states are reachable from any state in which all extender variables differ from their regulators.

It is worth observing that the elimination of the extender components from the extended network using the method described in Naldi et al. (2011) allows to recover the original network. The asynchronous dynamics of an extended network is thus an over-approximation of the original asynchronous dynamics. As consequence, If y is L-reachable from x, then it is also K-reachable for any $K \supset L$.

Example 14 Consider the Boolean network with three components $f(x_1,x_2,x_3) = (1, x_1 \wedge (x_2 \vee \neg x_3), x_2 \vee x_3)$. It has two fixed points, the states $a = (1,0,1)$ and $b = (1,1,1)$. Starting from $x = (0,0,0)$, in the asynchronous dynamics the first component can be updated to 1, followed by the second and the third component, to reach the fixed point $b = (1,1,1)$ (see Fig. 1c). No trajectory leads from (0, 0, 0) to the fixed point a, whereas a is clearly reachable from (0, 0, 0) in the most permissive semantics. The network can be extended to a cuttable network using extender variables $L = \{(2,2),(2,3),(3,2),(3,3)\}$. In the extended network, the state a is L-reachable from x, as can be seen for instance using Theorem 10.

In the following we compare in more detail the reachability properties of the original network and its cuttable extensions and relate the trap spaces of a Boolean network (V, f) to the trap spaces of its L-extension.

Observe that the image under ϵ of a subspace $[x,y] \subseteq \mathbb{B}^V$ is the subspace $\epsilon([x,y]) = [\epsilon(x),\epsilon(y)]$ with $\Delta(\epsilon(x),\epsilon(y)) = \Delta(x,y) \cup I'$ where I' is the subset of extender variables $\{(j,i) \in L$ such that $j \in \Delta(x,y)\}$. By extending the terminology from states to subspaces, we call subspaces of this form *canonical*.

Proposition 14 *Consider a Boolean network (V, f) and its L -extension* $(V \cup L, f^L)$.

(i) *If* $[x, y]$ *is a trap space for f, then* $\epsilon([x,y])$ *is a canonical trap space for* f^L. *If* $[x, y]$ *is the minimal trap space containing x, then* $\epsilon([x,y])$ *is the minimal trap space containing* $\epsilon(x)$.
(ii) *If* $[x',y']$ *is a trap space for* f^L, *then* $[\pi(x'),\pi(y')]$ *is a trap space for f and* $\Delta(\pi(x'),\pi(y')) = \Delta(x',y') \cap V$. *If* $[x',y']$ *is the minimal trap space containing* x', *then* $[\pi(x'),\pi(y')]$ *is the minimal trap space containing* $\pi(x')$.

Proof

(I) The fact that subspaces $\epsilon([x,y])$ and $[\pi(x'),\pi(y')]$ are trap spaces is a direct consequence of the definitions of f^L, ϵ and π.
(II) Suppose that $[x, y]$ is minimal, and consider a trap space $[w',z']$ contained in $[\epsilon(x),\epsilon(y)]$, that is, such that $\Delta(w',z') \subseteq \Delta(\epsilon(x),\epsilon(y))$. We have to show that $[w',z'] = [\epsilon(x),\epsilon(y)]$. By point (I), $[\pi(w'),\pi(z')]$ is a trap space contained in $[x, y]$, hence it coincides with $[x, y]$. As a consequence, $\Delta(\pi(w'),\pi(z')) = \Delta(w',z') \cap V = \Delta(x,y)$. Consider $(j,i) \in \Delta(\epsilon(x),\epsilon(y)) \cap L$, then $j \in \Delta(x,y) =$

$\Delta(\pi(w'), \pi(z'))$ by definition. Since $[w', z']$ is a trap space, by definition of f^L we have $(j, i) \in \Delta(w', z')$. Hence $\Delta(w', z') = \Delta(\epsilon(x), \epsilon(y))$, which concludes.

(III) Suppose now that $[x', y']$ is a minimal trap space for f^L; we show that $[\pi(x'), \pi(y')]$ is minimal. Consider a trap space $[z, t]$ contained in $[\pi(x'), \pi(y')]$. Then, by point (I), $\epsilon([z, t])$ is a trap space contained in $[x', y']$, hence coincides with $[x', y']$. As a consequence, their projections $\pi(\epsilon([z, t])) = [z, t]$ and $[\pi(x'), \pi(y')]$ are equal. □

The proposition states that all trap spaces in extended networks project to trap spaces for the original network, and any trap space in the original network gives at least one trap space in any extension. In addition, if y is a canonical state in an extended network, that is $y = \epsilon(x)$ for some x, then the minimal trap space containing y is the canonical extension of the minimal trap space containing x.

Clearly a Boolean network and its extensions do not necessarily have the same number of trap spaces. Multiple trap spaces in an extension can project to the same trap space in the original network. Take for instance the Boolean network $f(x_1) = x_1$ and its extension $f^L(x_1, x_2) = (x_2, x_1)$ with $L = (1, 1)$. The trap spaces 00 and 0★ for f^L project on the same trap space (the fixed point 0). On the other hand, the mapping between trap spaces described in the proposition defines a one-to-one correspondence between minimal trap spaces of a Boolean network and any of its extensions.

Corollary 15 *There is a one-to-one correspondence between the minimal minimal trap spaces of a Boolean network and the minimal trap spaces of any of its extensions.*

Remark 6 Minimal trap spaces in extended networks are always canonical. Every trap space t in an extended network contains the canonical trap space $\epsilon(\pi(t))$.

We now focus our study on cuttable extensions. As stated above, the full extension is always cuttable, but other cuttable extensions often exist in practice. Following the definition of cuttable networks, these more conservative cuttable extensions can be obtained by extending only interactions (i, j) such that $|T(i)| > 1$ and $|R(j)| > 1$ as well as one interaction for each cycle which remains unextended. The following properties build on the previous results obtained on cuttable networks and can be applied to any cuttable extension.

Proposition 16 *Let M be a Boolean network and L a subset of its interactions defining a cuttable extension.*

(i) *If there is a trajectory from x to y in the generalized asynchronous dynamics of M, then y is L -reachable from x.*

(ii) *Given a state x and t the minimal trap space containing x, all trap spaces contained in t are L -reachable from x.*

(iii) *There is a one-to-one correspondence between the minimal trap spaces of M and the attractors in the asynchronous dynamics of its L -extension.*

Proof

(i) It is sufficient to show that, if $\bar{x}^J$ is a successor of x in the generalized asynchronous dynamics of M, then $\bar{x}^J$ is L-reachable from x. By definition of extended network we have, for all $i \in J$, $\mathcal{E}(f, L)_i(\epsilon(x)) = f_i(x) \neq x_i = \epsilon_i(x)$, and $\overline{\epsilon(x)}^J$ is a successor of $\epsilon(x)$ in the generalized asynchronous dynamics of the extended network. By Corollary 7(ii), $\overline{\epsilon(x)}^J$ is reachable from $\epsilon(x)$ in the asynchronous dynamics of the extended network. Since $\overline{\epsilon(x)}^J$ and $\epsilon(\bar{x}^J)$ coincide on the core variables and $\epsilon(\bar{x}^J)$ is canonical, $\epsilon(\bar{x}^J)$ can be reached from $\overline{\epsilon(x)}^J$. Combining the two paths we have that $\epsilon(\bar{x}^J)$ is reachable from $\epsilon(x)$.

(ii) Consider a trap space t' contained in t. By Proposition 14, $\epsilon(t')$ is a trap space contained in $\epsilon(t)$, and $\epsilon(t)$ is the minimal trap space containing $\epsilon(x)$. Theorem 10 then gives that $\epsilon(t')$ is reachable from $\epsilon(x)$ in the extended network, that is, there exists $y \in \epsilon(t')$ such that there is a path from $\epsilon(x)$ to y in the asynchronous dynamics of the extended network. In addition, we can assume that y is canonical, that is, $\epsilon(\pi(y)) = y$. Then $\pi(y)$ is in t' is L-reachable from from x.

(iii) Consequence of Theorem 11 and Corollary 15. □

Example 15 A cuttable extension can be obtained from the TCR signaling network of Fig. 3 by adding 23 linear variables extending the interaction highlighted in bold. For instance all regulations with target PAGCsk need to be considered in the extension as it has several regulators and they all have multiple targets. On the contrary, the regulation of LCK by PAGCsk does not need to be extended as PAGCsk has a single target and all of its incoming regulations are already considered.

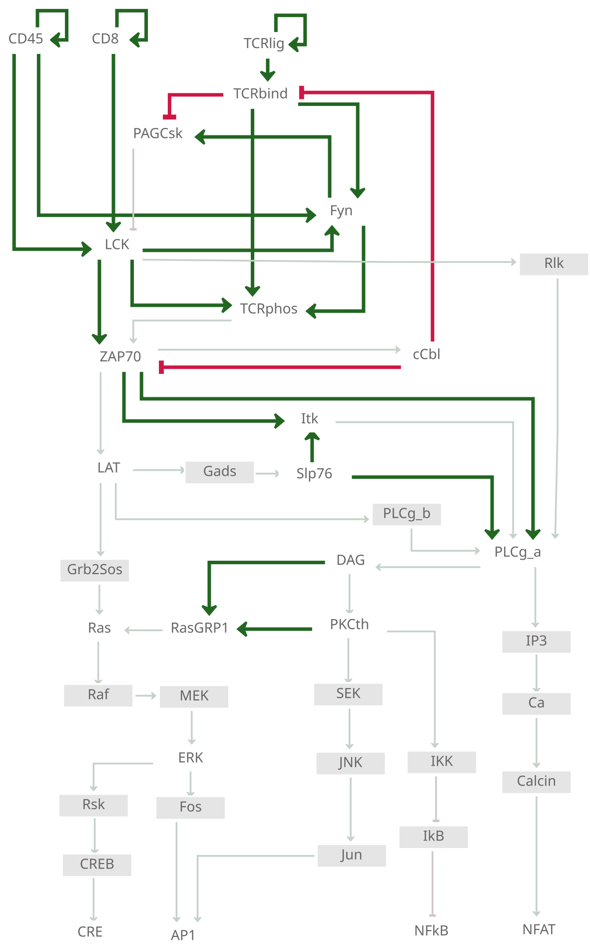

Fig. 3 Cuttable extension of a biological model. The interaction graph of the TCR signaling network studied in Klamt et al. (2006) has 40 components and 58 interactions. Grey nodes denote existing linear components. Bold edges need to be extended to obtain a cuttable extension

5.2 Relation to single threshold refinements

Multi-valued networks are commonly used to refine the behaviour of some components of a Boolean network. They can account for some semi-quantitative knowledge, for instance by tracking different amounts of a component that are required to affect its different targets, or by encoding the existence of some specific condition leading to a higher production or a higher activity level for some target. To account for all these effects, multi-valued refinements can take many forms and involve complex modifications to the logical rules (Chaouiya et al. 2003). Here we introduce *single threshold* networks, a subset of multi-valued networks that adds different thresholds to the interactions but retains the same logical rules as the Boolean network. Such refinements are solely defined by a Boolean network and a mapping associating a single multi-valued threshold to each interaction of the network.

We start by setting some notation and definitions. Given a Boolean network $M=(V,f)$ with $V=\{1,\ldots,n\}$, we call any $\tau: V^2 \to \mathbb{N}^*$ a *threshold map* for M. For each $i\in V$, we then define the value m^i and the mapping $\Omega^i: \mathbb{N}^V \to \mathbb{B}^V$ such that:

$$m^i = \max(\{1\} \cup \{\tau(i,j) \mid j \in T(i)\}),$$
$$\Omega^i(x)_j = \mathbb{1}(x_j \geq \tau(j,i)) \quad \text{for each } j \in V,$$

where $\mathbb{1}(P)$ equals 1 when the statement P is true, and 0 otherwise. We call $\aleph = \prod_{i\in V}[0,m^i]$ the *multi-valued space* of (M,τ). For each component i, we denote by e^i the element of $\aleph$ with component i equal to 1 and all other components equal to 0. In addition, we define the mapping $\rho: \mathbb{B}^V \to \aleph$ such that for each component $i\in V$, $\rho(x)_i = m^i \cdot x_i$.

Definition 6 Given a Boolean network $M=(V,f)$ and a threshold map τ for M, the function

$$\mathcal{R}(f,\tau): \aleph \to \aleph$$
$$\mathcal{R}(f,\tau)_i = \rho_i \circ f \circ \Omega^i \quad \text{for all } i \in V$$

is called the τ-*refinement* of M. The multi-valued network $\mathcal{M} = (\aleph, \mathcal{R}(f,\tau))$ is a *single threshold refinement* of M.

As is customary for multi-valued networks we consider dynamics that allow for asynchronous stepwise transitions that point in the direction defined by the multi-valued function. That is, we define the asynchronous dynamics of $\mathcal{M}$ as the graph with vertex set $\aleph$ and edge set $\{(x, x+\varepsilon \mathrm{e}^i) \mid x\in\aleph,\ i \in \Delta(x,\mathcal{R}(f,\tau)(x)),\quad \varepsilon = \mathrm{sign}(\mathcal{R}(f,\tau)_i(x) - x_i)\}$.

Proposition 17 *Let M be a Boolean network and τ a threshold map for M. If there exists a transition $x \to \bar{x}^i$ in the asynchronous dynamics of M and there is no transition $\bar{x}^i \to x$, then there is a trajectory from $\rho(x)$ to $\rho(\bar{x}^i)$ in the asynchronous dynamics of the τ -refinement of M.*

Proof Define $y^\sigma = \rho(x) + \varepsilon\sigma \mathrm{e}^i$ for all $\sigma = 0,\ldots,m^i$, where $\varepsilon = \mathrm{sign}(\mathcal{R}(f,\tau)_i(x) - x_i)$. We have $y^0 = \rho(x)$ and $y^{m^i} = \rho(\bar{x}^i)$. In addition, $\Omega^i(y^\sigma)_j = x_j$ for all $j\neq i$, and since $f_i(x) = f_i(\bar{x}^i)$ we get $\mathcal{R}(f,\tau)_i(y^\sigma) = m^i \cdot f_i(x)$ for all σ, and there is a transition $y^\sigma \to y^{\sigma+1}$ for all $\sigma = 0,\ldots,m^i-1$. □

The interaction graph G of a Boolean network $M=(V,f)$ can be endowed with a label function $S: E \to \mathcal{P}(\{-1,1\})$ that assigns signs to edges. For an edge $(j,\ i)$ in E and $s\in\{-1,1\}$, we have $s\in S((j,i))$ if there exists a state $x\in\mathbb{B}^V$ such that $(f_i(\bar{x}^j) - f_i(x))(\bar{x}^j_j - x_j) = s$. Proposition 17 then gives the following corollary.

Corollary 18 *Let $M=(V,f)$ be a Boolean network and suppose that the interaction graph of f has no loops with* *negative sign. If there is a path from x to y in the asynchronous dynamics, then there is a path from $\rho(x)$ to $\rho(y)$ in the asynchronous dynamics of all single threshold refinements of M.*

For some single threshold refinements of Boolean networks with negative loops in the interaction graph, the asynchronous dynamics can contain oscillations at intermediate levels and fail to capture the Boolean dynamics.

Example 16 Consider the Boolean network $(\{1,2\},f)$ with $f(x_1,x_2) = (\neg x_1, x_1)$. The map $\tau : \{1,2\}^2 \to \mathbb{N}^*$ defined by $\tau(1,1) = 1$, $\tau(1,2) = 2$, $\tau(2,1) = \tau(2,2) = 0$ is a threshold map for *f*. The associated τ-refinement is given by $\aleph = \{0,1,2\} \times \{0,1\}$, $\mathcal{R}(f,\tau)_1(y_1,y_2) = 2f_1(\mathbb{1}(y_1 \geq 1), 1)$, $\mathcal{R}(f,\tau)_2(y_1,y_2) = f_2(\mathbb{1}(y_1 \geq 2), 1)$, so that (0, 0) and (0, 1) are mapped to (2, 0), (1, 0) and (1, 1) are mapped to (0, 0), and (2, 0) and (2, 1) are mapped to (0, 1). There is a transition from (0, 0) to (1, 0) in the Boolean asynchronous dynamics, but there is no trajectory from $\rho(0,0) = (0,0)$ to $\rho(1,0) = (2,0)$ in the multi-valued asynchronous dynamics.

We now want to study the ability of linear extensions of a Boolean network *f* to capture behaviours of single threshold refinements of *f*. We will show that to each path found in single threshold refinements corresponds a path in the asynchronous dynamics of a full extension. To formalize this, we will use the notion of *L*-reachability that we introduced in the previous section, and we will need an auxiliary function Γ which associates, to each multi-valued state of a refinement, a set of Boolean states in the extended network (to be more precise, a subspace of the state space of the extended network).

The mapping defined by Γ works as follows. In the multi-valued space, the multi-valued variable $j \in V$ determines possible values for component *j* in $\mathbb{B}^{V \cup L}$ as well as for linear variables that have *j* has their unique regulator. For a variable (*j*, *k*) in *L*, in order to determine whether a component is "on" or "off", the value of its regulator is compared against the associated threshold $\tau(j,k)$. For a variable *j* outside *L*, extreme values, 0 and m^j, are mapped to 0 and 1 respectively, whereas intermediate values leave component *j* in the extended network "undefined" (they are mapped to $\star$). This flexibility allows us to easily derive the existence of the desired paths in the dynamics of the extended network (Corollary 20) by combining paths corresponding to transitions in single threshold refinements (Proposition 19).

Definition 7 Let $M = (V,f)$ be a Boolean network, $L \subseteq E$ a subset of its interactions, $M^L = (V \cup L, \mathcal{E}(f,L))$ the associated extension. Let τ be a threshold map for *M*, with $\aleph$ the associated multi-valued space. We define the mapping $\Gamma : \aleph \to \{0,1,\star\}^{V \cup L}$ as follows:

$$\Gamma(x)_i = \begin{cases} 0 & \text{if } i \in V \text{ and } x_i = 0, \\ \star & \text{if } i \in V \text{ and } 0 < x_i < m^i, \\ 1 & \text{if } i \in V \text{ and } x_i = m^i, \\ \mathbb{1}(x_j \geq \tau(j,k)) & \text{if } i = (j,k) \in L. \end{cases}$$

for all $x \in \aleph$ and $i \in V \cup L$.

If $x \in \mathbb{B}^V$ is a state of the Boolean network, then $\Gamma(\rho(x)) = \epsilon(x)$.

Proposition 19 *Let $M = (V,f)$ be a Boolean network, $(\aleph, \mathcal{R}(f,\tau))$ the single threshold refinement of M associated to a threshold map τ and $(V \cup E, \mathcal{E}(f,E))$ the full extension of M. If there is a transition $x \to y$ in the asynchronous dynamics of $\mathcal{R}(f,\tau)$, then for each state $z \in \Gamma(x)$ there is a geodesic from z to at least one state $z' \in \Gamma(y)$ in the asynchronous dynamics of $\mathcal{E}(f,E)$.*

Proof Let *i* be the only component such that $x_i \neq y_i$. We call $v = f_i(\Omega^i(x))$ the Boolean target value of *i* at $\Omega^i(x)$. We have $x_i \neq \mathcal{R}(f,\tau)_i(x) = m^i \cdot v$. Take a state $z \in \Gamma(x)$. For each regulator *j* of *i* we have $\pi^i(z)_j = z_{(j,i)} = \Omega^i(x)_j$, hence $\mathcal{E}(f,E)_i(z) = f_i(\pi^i(z)) = f_i(\Omega^i(x)) = v$.

By definition of Γ, $\Gamma_i(y) \in \{v,\star\}$. Note that $\Gamma(x)$ and $\Gamma(y)$ can differ only in component *i* and in linear components associated to edges with regulator *i*. Call $w \in \Gamma(y) \in \mathbb{B}^{V \cup E}$ the unique state such that $w_i = v$ and $\Delta(z,w) \subseteq \{i\} \cup \{(i,k) \mid k \in T(i)\}$. We will show that there is a geodesic from *z* to *w*.

As the extended network is a full extension, all targets of *i* in the interaction graph of $\mathcal{E}(f,E)$ are in *E*. Let $U = \Delta(z,w) \setminus \{i\} = \Delta(z,w) \cap E$ be the set of targets of *i* that differ in *w* and *z*. For each $e = (i,k) \in U$, we have $\mathbb{1}(x_i \geq \tau(i,k)) = z_e \neq w_e = \mathbb{1}(y_i \geq \tau(i,k)) = v$.

If $z_i = v$ then there is a geodesic from *z* to *w* that consists in updating all components of *U* (this is possible in any order). If $z_i \neq v$ then since $\mathcal{E}(f,E)_i(z) = v$ there is a transition $z \to \bar{z}^i$, followed by a similar geodesic from $\bar{z}^i$ to *w*. □

Corollary 20 *Consider a Boolean network (V, f) and x, y Boolean states. If there exists a threshold map τ such that $\rho(y)$ is reachable from $\rho(x)$ in the asynchronous dynamics of $\mathcal{R}(f,\tau)$, then y is E -reachable from x.*

Note that in Proposition 19 and Corollary 20, we only considered the full extension. Whether the conclusions hold for any cuttable extension remains an open question.

Example 17 Consider the Boolean network in Example 14, and a single threshold multi-valued refinement that "separates" the thresholds of the regulations $2 \to 2$ and $2 \to 3$. That is, take $m^1 = 1$, $m^2 = 2$, $m^3 = 1$, $\tau(1,2) =$

$\tau(2,3) = \tau(3,2) = \tau(3,3) = 1$ and $\tau(2,2) = 2$. The resulting τ-refinement is $\mathcal{R}(f,\tau) = (1, 2(\mathbb{1}(x_1 \geq 1) \wedge (\mathbb{1}(x_2 \geq 2) \vee \mathbb{1}(x_3 < 1))), \mathbb{1}(x_2 \geq 1) \wedge \mathbb{1}(x_3 \geq 1))$.

The refinement admits a trajectory from (0, 0, 0) to the fixed point (1, 0, 1). In fact, a linear extension of the Boolean network with extender variable set $L = \{(2,2)\}$ is also sufficient to give the L-reachability of (1, 0, 1) from (0, 0, 0) (see Fig. 1c).

6 Discussion

To reflect the lack of kinetic knowledge often associated with biological networks, the classical asynchronous semantics explores all possible alternative trajectories where a single component is updated in each transition. The generalized asynchronous semantics accounts for possible partial or total synchronism in updates. The binary nature of activity levels on the other hand implies that a change of the activity level of a single component simultaneously affects all its target components. In many networks, the effect of a component on different targets involves different mechanisms with their own kinetics and even sometimes different implicit intermediates. In case of competition (such as the inconsistent feedback loop in Fig. 1a), the classical semantics then fail to capture some plausible behaviours. Multi-valued networks could be used to define separate thresholds for different targets, but would require either additional knowledge for all interactions or the identification of some key interactions that would benefit from a refinement. The most permissive semantics uses transitory states to address this issue and reproduces the behaviour of all multi-valued refinements, but also introduces undesired non monotonic behaviours. For example, a component in the increasing state can act in succession as inactive, then active, then inactive again for one of its targets as illustrated in Fig. 1b. While such behaviours could be interpreted as stochastic effects in the neighbourhood of an activation threshold, they can often be considered as artefacts. Here, we focused on single threshold refinements, a small subset of multi-valued refinements that enable threshold separation while preserving the original Boolean functions (thus without introducing non monotonic behaviours). The extension of individual interactions with linear components can be used to emulate such refinements in absence of knowledge on the threshold values and within the established framework of asynchronous Boolean networks. While the cuttable extended semantics is not a minimal abstraction for single threshold refinements, it provides a stricter over-approximation than the most permissive semantics for this class of networks, which can be particularly interesting from a modeling perspective.

As a tool to study asynchronous trajectories we introduced implicant maps representing dependencies and conflicts controlling the possible change of value of the components compared to a specific initial state. These implicant maps correspond to classes of subgraphs in the implicant graph used for the identification of trap spaces (stable motifs, see Zañudo and Albert 2013) or equivalently in the Petri net unfolding of the Boolean network (Chaouiya et al. 2011). We say that an implicant map is weakly consistent if it describes a set of satisfiable (complete and non-circular) dependencies. In absence of any weakly consistent map containing a given component, we know that there is no trajectory (in any semantics) in which the value of this component can be modified. This strong requirement is consistent with our observation that the maximal weakly consistent maps correspond to the smallest trap spaces containing the initial state. This weak consistency solely relies on dependencies and ignores the competition between components. In permissive trajectories this limitation is ignored and all components included in a weakly consistent map can be updated in a geodesic (following a partial order defined by the dependencies). However these competitions can play a role in asynchronous trajectories, where some of these components can only be updated after much longer trajectories, if ever. A weakly consistent implicant map is strongly consistent in absence of competition between its components. This stronger consistency property is both necessary and sufficient for the existence of asynchronous geodesics.

As the direct requirements and competitions described by implicant maps are associated to interactions in the interaction graph, the consistency constraints correspond to undirected cycles in the interaction graph. We further observed that a linear component mirroring its unique regulator in the initial state can be used to relax such competitions. This led us to study the dynamical properties of cuttable networks, a structural class of Boolean networks in which a set of linear components cover all feedback loops and paths from any component with multiple targets to any component with multiple regulators. Our observations suggest that these two structural conditions correspond to different types of competitions. On one hand, the linear extension of feedback loops seems to be associated to the synchronized update of multiple components, as illustrated in Fig. 1d. It is thus required and could be sufficient to reproduce the generalized asynchronous trajectories. On the other hand, the linear extension of paths connecting a component with multiple targets to a component with multiple regulators could be related to threshold separation in feedforward loops. We observed strong similarities between the trajectories recovered through the extension of feedforward loops and in single threshold refinements as illustrated in Fig. 1a, c. These two associations are consistent with the fact that the extended dynamics reproduces the reachability properties obtained in both the generalized asynchronous and all single threshold refinements. Further work is needed to clarify the role of feedback loops, feedforward loops, and other paths from components with multiple targets to components with multiple regulators in the dynamical properties of cuttable networks to elucidate whether the structural conditions for linear cuts could then be further generalized.

We have implemented the linear extension of Boolean networks in the bioLQM software (Naldi 2018), enabling the use of the extended semantics in existing software tools supporting the classical asynchronous semantics. Note that efficient analysis based on trap spaces does not require this explicit extension and can be performed directly on the original Boolean networks using existing implementations of trap spaces identification in PyBoolNet (Klarner et al. 2017), BioLQM or Trappist (Trinh et al. 2022).

As shown by Klarner et al. (2015), prime implicants provide a compact and complete representation of the implicant graph enabling the identification of sets of implicants that cooperatively define a trap space as the solutions of a constraint solving problem. We plan to adapt this approach to the identification of implicant maps with the desired consistency level. The identification of strongly-consistent maps can be used as a proof of reachability in the asynchronous semantics, while the identification of weakly consistent maps can be used to pinpoint specific competitions that need to be relaxed to enable this reachability. Beyond the general question of reachability, this approach would provide valuable hints to assess the biological relevance of the corresponding extended trajectories. Note that this type of reasoning can only be used to formally validate a reachability property: if the competitions can not be realistically relaxed, then more complex trajectories to the target of interest may still exist.

7 Conclusion

In this paper we study the reachability properties of dynamical Boolean networks, and in particular the reachability of a subspace from a specific initial state. This question is known to be PSPACE-complete in the classical asynchronous semantics, however abstract interpretation approaches provide efficient solutions in some cases (Paulevé et al. 2012, 2020). Furthermore, this problem is polynomial for monotonic networks in the recently proposed most permissive semantics (Paulevé et al. 2020). This novel semantics extends the classical asynchronous semantics by adding intermediate activity levels explicitly accounting for the absence of information on the regulation thresholds. This approach enables the simulation of relevant behaviours missed by the standard asynchronous dynamics. The most permissive semantics can, on the other hand, also introduce some artefactual behaviours and should thus be considered as an over-approximation. This work starts with the characterisation of different structural conditions for individual transitions in asynchronous and permissive trajectories and leads to the identification of a class of Boolean networks and initial states for which these semantics have the same geodesics. These networks have a simple structural characterization: they are networks whose interaction graph admits a *linear cut*. We could show that trap spaces (also called stable motifs or symbolic steady states, see Zañudo and Albert 2013; Klarner et al. 2014) always provide a precise characterization of all attractors in cuttable networks, and that their reachability solely depends on the minimal trap space containing the initial state (Fig. 2). These results are strong improvements compared to the general case where trap spaces lack such formal guarantees, even if they are often considered as good estimators in practice. These results are similar to the properties of the most permissive dynamics but here they do not rely on intermediate activity levels that could induce known artefactual behaviours.

We then proposed an extended semantics based on linear extensions of Boolean networks. This type of extension can be interpreted as the explicit representation of hidden delays or threshold effects, and thus carries a natural biological justification. As trap spaces of the original network are also trap spaces of their extensions, the properties of cuttable networks (reachability of trap spaces and configuration of attractors) can then be applied directly to any Boolean network without explicitly constructing a cuttable extension. The reachability properties of this extended semantics provide an interesting middle ground between the asynchronous semantics and the most permissive semantics, as it recovers realistic trajectories missing in the former and excludes some artefactual behaviours of the latter (see Fig. 4). The reachability of trap spaces in the cuttable extension semantics has the same polynomial complexity as in the most permissive; however, the reachability of transient subspaces remains to be investigated. It is currently unclear if all permissive trajectories which are not captured by this new semantics are associated to non-monotonicity (and could be considered as artefacts) or if some relevant trajectories (to transient states) might also missing. Similarly, while the most permissive semantics capture all possible behaviours of multi-valued refinements, the ability of our extended semantics to reproduce behaviours emerging in multi-valued refinements has been only partially explored. We have shown that refinements that rely on a unique threshold per regulation can be captured by full extensions; however this

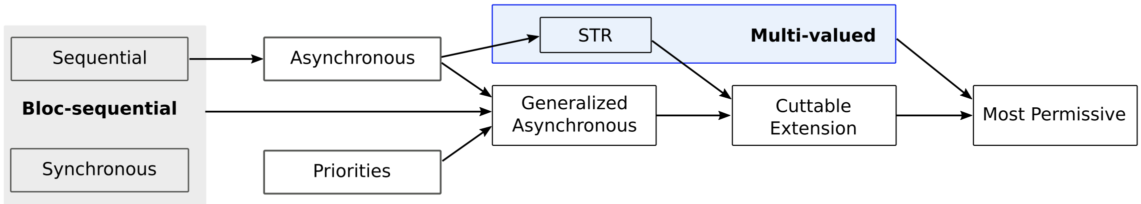

Fig. 4 Reachability properties across updating semantics. Boxes represent updating semantics and arrows between them indicate that the target semantics is an over-approximation of the source semantics. The gray area on the left groups classical deterministic semantics, while all others are non-deterministic. STR stands for single threshold refinement (Definition 6), and the blue area denotes the asynchronous semantics of all multi-valued refinements

condition does not fully characterize the emerging behaviours.

The strength of Boolean networks lies in their simple, parameter-free formulation. However, their ability to deal with lack of detailed kinetic information is also at the core of their intrinsic limitations. Although the parameter uncertainty can partially be encoded by resorting to non-deterministic semantics, many potential fine-grained behaviours that depend on specific parameter scenarios are inevitably inaccessible when relying to logical rules alone. The most permissive semantics provide an important step to ensure that all possible parameters are indeed captured, and can thus be used to formally rule out reachability properties which are structurally impossible for any set of parameters. However, it also increases the number of artefactual trajectories in the system. Implicant maps provide the groundwork to formally identify trajectories which remain realistic for any set of parameters or for parameters matching well-characterized conditions. These maps can be constructed for direct trajectories (geodesics) in the permissive or extended semantics as shown here and could be naturally extended to trajectories where all components are updated at most twice, which can be required for the reachability of some trap spaces. However, it would not scale to arbitrarily complex trajectories, which remain in a gray area. We could imagine combining these approaches to annotate any reachability property as formally impossible, unlikely, realistic or formally guaranteed.

Acknowledgements We thank Heike Siebert for insightful discussions.

Author contributions AN and ET conceived and developed the project. All authors expanded and formulated the theory, contributed to the manuscript, read and approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. AN was supported by the Deutsche Forschungsgemeinschaft (GRK 1772) and French National Research Agency (ANR-MOST project ANR-16-CE18-0029 “BIOPSY”). AR was supported by the French National Research Agency (Young Researcher project ANR-18-CE40-0002-01 “FANs”). ET was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy - The Berlin Mathematics Research Center MATH+ (EXC-2046/1, project ID: 390685689).

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

Béal J, Pantolini L, Noël V et al (2021) Personalized logical models to investigate cancer response to BRAF treatments in melanomas and colorectal cancers. PLoS Comput Biol 17(1):e1007-900. https://doi.org/10.1371/journal.pcbi.1007900

Bonzanni N, Garg A, Feenstra KA et al (2013) Hard-wired heterogeneity in blood stem cells revealed using a dynamic regulatory network model. Bioinformatics 29(13):i80–i88. https://doi.org/10.1093/bioinformatics/btt243

Chaouiya C, Remy E, Mossé B et al (2003) Qualitative analysis of regulatory graphs: a computational tool based on a discrete formal framework. Positive systems. Springer, Berlin, pp 119–126. https://doi.org/10.1007/978-3-540-44928-7_17

Chaouiya C, Naldi A, Remy E et al (2011) Petri net representation of multi-valued logical regulatory graphs. Nat Comput 10(2):727–750. https://doi.org/10.1007/s11047-010-9178-0

Chevalier S, Froidevaux C, Paulevé L et al (2019) Synthesis of Boolean networks from biological dynamical constraints using answer-set programming. In: 2019 IEEE 31st international conference on tools with artificial intelligence (ICTAI). IEEE, pp 34–41. https://doi.org/10.1109/ICTAI.2019.00014

Cohen DPA, Martignetti L, Robine S et al (2015) Mathematical modelling of molecular pathways enabling tumour cell invasion and migration. PLOS Comput Biol 11(11):e1004571. https://doi.org/10.1371/journal.pcbi.1004571

Collombet S, van Oevelen C, Sardina Ortega JL et al (2017) Logical modeling of lymphoid and myeloid cell specification and transdifferentiation. Proc Natl Acad Sci 114(23):5792–5799. https://doi.org/10.1073/pnas.1610622114

Dubrova E, Teslenko M (2011) A sat-based algorithm for finding attractors in synchronous Boolean networks. TCBB 8(5):1393–1399. https://doi.org/10.1109/TCBB.2010.20

Fauré A, Naldi A, Chaouiya C et al (2006) Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. Bioinformatics 22(14):e124–e131. https://doi.org/10.1093/bioinformatics/btl210

Kauffman SA (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. J Theor Biol 22(3):437–467. https://doi.org/10.1016/0022-5193(69)90015-0

Klamt S, Saez-Rodriguez J, Lindquist JA et al (2006) A methodology for the structural and functional analysis of signaling and regulatory networks. BMC Bioinformatics 7:1–26. https://doi.org/10.1186/1471-2105-7-56

Klarner H, Bockmayr A, Siebert H (2014) Computing symbolic steady states of Boolean networks. In: Cellular Automata, pp 561–70. https://doi.org/10.1007/978-3-319-11520-7_59

Klarner H, Bockmayr A, Siebert H (2015) Computing maximal and minimal trap spaces of Boolean networks. Nat Comput 14(4):535–544. https://doi.org/10.1007/s11047-015-9520-7

Klarner H, Streck A, Siebert H (2017) PyBoolNet: a python package for the generation, analysis and visualization of Boolean networks. Bioinformatics 33(5):770–772. https://doi.org/10.1093/bioinformatics/btw682

Naldi A (2018) BioLQM: a Java toolkit for the manipulation and conversion of logical qualitative models of biological networks. Front Physiol 9:1605. https://doi.org/10.3389/fphys.2018.01605

Naldi A, Thieffry D, Chaouiya C (2007) Decision diagrams for the representation and analysis of logical models of genetic networks. In: Computational methods in systems biology, pp 233–247. https://doi.org/10.1007/978-3-540-75140-3_16

Naldi A, Remy E, Thieffry D et al (2011) Dynamically consistent reduction of logical regulatory graphs. Theoret Comput Sci 412(21):2207–2218. https://doi.org/10.1016/j.tcs.2010.10.021

Noual M, Sené S (2018) Synchronism versus asynchronism in monotonic Boolean automata networks. Nat Comput 17:393–402. https://doi.org/10.1007/s11047-016-9608-8

Paulevé L, Magnin M, Roux O (2012) Static analysis of biological regulatory networks dynamics using abstract interpretation. Math Struct Comput Sci 22(4):651–685. https://doi.org/10.1017/S0960129511000739

Paulevé L, Kolčák J, Chatain T et al (2020) Reconciling qualitative, abstract, and scalable modeling of biological networks. Nat Commun. https://doi.org/10.1038/s41467-020-18112-5

Remy E, Mossé B, Chaouiya C et al (2003) A description of dynamical graphs associated to elementary regulatory circuits. Bioinformatics 19(suppl_2):ii172–ii178. https://doi.org/10.1093/bioinformatics/btg1075

Richard A (2018) Fixed points and connections between positive and negative cycles in Boolean networks. Discret Appl Math 243:1–10. https://doi.org/10.1016/j.dam.2017.12.037

Robert F (1986) Discrete iterations, vol 6. Springer series in computational mathematics. Springer, Berlin. https://doi.org/10.1007/978-3-642-61607-5

Sizek H, Hamel A, Deritei D et al (2019) Boolean model of growth signaling, cell cycle and apoptosis predicts the molecular mechanism of aberrant cell cycle progression driven by hyperactive PI3K. PLOS Comput Biol 15(3):e1006402. https://doi.org/10.1371/journal.pcbi.1006402

Thomas R (1973) Boolean formalization of genetic control circuits. J Theor Biol 42(3):563–585. https://doi.org/10.1016/0022-5193(73)90247-6

Trinh VG, Benhamou B, Hiraishi K et al (2022) Minimal trap spaces of logical models are maximal siphons of their petri net encoding. In: International conference on computational methods in systems biology. Springer, Berlin, pp 158–176. https://doi.org/10.1007/978-3-031-15034-0_8

Zañudo JGT, Albert R (2013) An effective network reduction approach to find the dynamical repertoire of discrete dynamic networks. Chaos 23(2):025111. https://doi.org/10.1063/1.4809777