# Compositional Techniques for Asynchronous Boolean Networks

Maram Alshahrani[1,2] [a] and Jason Steggles[1] [b]
[1]School of Computing, Newcastle University, Newcastle upon Tyne, U.K.
[2]Faculty of Computer Science, King Khalid University, Abha, Saudi Arabia

Keywords: Asynchronous Boolean Networks, Model Composition, Behaviour Preservation, Attractor Identification.

Abstract: Asynchronous Boolean networks are an important qualitative modelling approach for analysing and engineering biological systems. However, their practical application is limited by the state space explosion problem and lack of engineering tools. To help address these limitations we develop new compositional techniques for constructing and analysing asynchronous Boolean networks based on the idea of merging entities using Boolean operators. We propose a novel *asynchronous interference state graph* to model the interference that occurs in a composition and develop a range of important new asynchronous compositional techniques for analysing behavioural preservation and identifying point attractors.

## 1 INTRODUCTION

*Boolean networks* (Kauffman, 1969) are a widely studied qualitative modelling approach (Barbuti et al., 2020) that has been successfully applied to a range of biological systems (for example, see (Pandey et al., 2010; Dahlhaus et al., 2016)). A Boolean network abstractly represents the state of regulatory entities as Boolean values and provides update rules to capture how these entities interact. They can provide key insights into the behaviour of a regulatory system by considering the *attractors* (state cycles) they exhibit (for example, see (Huang and Ingber, 2000; Saadatpour et al., 2011)). The global state of a Boolean network can be updated either *synchronously* (all entities update their state simultaneously) or *asynchronously* (entities update their state independently and non-deterministically) (Schwab et al., 2020). While the synchronous update scheme provides more tractable analysis, the asynchronous update scheme is generally seen to allow more realistic dynamic behaviour (Barbuti et al., 2020).

The practical application of Boolean networks is limited by the *state space explosion problem* (Groote et al., 2015) (exponential growth of state space) and by the lack of engineering techniques. In this paper we make an important contribution to supporting the practical application of asynchronous Boolean networks by developing formal techniques for their compositional construction and analysis. We take as our starting point an approach for composing synchronous Boolean networks based on *merging entities between submodels using Boolean operators* (Alkhudhayr and Steggles, 2019; Abdulrahman and Steggles, 2023). Adapting this approach to the asynchronous setting involves dealing with the fundamental semantic differences between synchronous and asynchronous updating schemes and has led to a range of important new insights and results.

We begin by developing a new form of *interference state graph* (Alkhudhayr and Steggles, 2019) for the asynchronous setting which captures the interference that can occur to a Boolean network's asynchronous behaviour in a composition. We prove this interference state graph bounds the asynchronous behaviour possible for a submodel in a composition and then develop key results about the preservation of a submodel's behaviour in a composition.

We further strengthen our asynchronous compositional framework by developing new compositional techniques for identifying *point attractors* (Hopfensitz et al., 2013) (global states with no outgoing transitions) in an asynchronous composition. We formulate an approach based on identifying and combining potential self-loops in the submodels, taking account of the impact that interference can have. The results here are promising and importantly provide a foundation for developing general techniques and tools for compositionally identifying cyclic and complex attractors.

[a] https://orcid.org/0009-0005-0793-0504
[b] https://orcid.org/0000-0001-9174-5531



Alshahrani, M. and Steggles, J.
Compositional Techniques for Asynchronous Boolean Networks.
DOI: 10.5220/0012324300003657

In *Proceedings of the 17th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2024) - Volume 1*, pages 429-437
ISBN: 978-989-758-688-0; ISSN: 2184-4305

# 2 BACKGROUND

## 2.1 Boolean Networks

A *Boolean network* (Kauffman, 1969) is a qualitative modelling approach which consists of a set of regulatory entities that have a Boolean state. Each entity has a next-state update function that defines its behaviour based on the state of its related entities.

**Definition 1.** *A Boolean network $\mathcal{BN}$ is a tuple $\mathcal{BN} = (G, N, F)$ where:*
*i) $G = \{g_1, g_2, ..., g_n\}$ is a non-empty, finite set of entities;*
*ii) $N = (N(g_1), N(g_2), ..., N(g_n))$ is a tuple of neighbourhoods, such that $N(g_i) \subseteq G$ is the neighbourhood of $g_i$;*
*iii) $F = (F(g_1), F(g_2), ..., F(g_n))$ is a tuple of next-state functions, where $F(g_i) : \mathbb{B}^{|N(g_i)|} \to \mathbb{B}$ defines the next state of $g_i$.*

We define a global state of a Boolean network $\mathcal{BN}$ to be a function $S : G \to \mathbb{B}$, where $S(g)$ represents the state of entity $g \in G$, and we let $S_{\mathcal{BN}} = [G \to \mathbb{B}]$ represent the set of all global states. If a Boolean network has $n \in \mathbb{N}$ entities then $S_{\mathcal{BN}}$ will contain $2^n$ states and this indicates that Boolean networks are impacted by the *state space explosion problem* (Groote et al., 2015). Given a global state $S \in S_{\mathcal{BN}}$ and a subset of entities $X \subseteq G$ we define the projection of $S$ over $X$ to be the function $S[X] : X \to \mathbb{B}$, where $S[X](g) = S[g]$, for any $g \in X$. For any $S \in S_{\mathcal{BN}}$, $g \in G$ and $b \in \mathbb{B}$ we let $S[g \to b]$ denote the state update which results in a new global state where $S[g \to b](h) = S(h)$, for all $h \in G$, $h \neq g$, and $S[g \to b](g) = b$.

A range of update semantics exist for Boolean networks and two key approaches are: *synchronous*, where all entities update their state simultaneously; and *asynchronous*, where each entity updates its state independently and non-deterministically (Schwab et al., 2020). The asynchronous semantics can be seen to capture more realistic dynamic behaviour (Barbuti et al., 2020) and we focus on this.

**Definition 2.** *Given global states $S, S' \in S_{\mathcal{BN}}$, we say $S \xrightarrow{\mathcal{BN}} S'$ is an asynchronous update step iff exists $g \in G$ such that $S'(g) = \neg S(g) = F(g)(S[N(g)])$ and $S(h) = S'(h)$, for all $h \in G$, $h \neq g$. We let $U_{\mathcal{BN}}(S) = \{S' \mid S' \in S_{\mathcal{BN}}$, and $S \xrightarrow{\mathcal{BN}} S'\}$. The complete asynchronous behaviour of a Boolean network is concisely captured by the (asynchronous) state graph $SG(\mathcal{BN}) = (S_{\mathcal{BN}}, \xrightarrow{\mathcal{BN}})$.*

For illustrative examples, consider the Boolean networks presented in Figure 1.

A *path* in the asynchronous state graph $SG(\mathcal{BN})$ represents one possible sequence of behaviour and can be finite or infinite (this contrasts with the synchronous setting where all paths are infinite (Kauffman et al., 1993)). We let $Path(SG(\mathcal{BN}))$ denote the set of all paths over $SG(\mathcal{BN})$. Given $\alpha \in Path(SG(\mathcal{BN}))$ we let $SD(\alpha)$ represent the step domain of $\alpha$, where $SD(\alpha) = \{0, \ldots, k-1\}$ if $\alpha : \{0, \ldots, k\} \to S_{\mathcal{BN}}$ is a finite path and $SD(\alpha) = \mathbb{N}$ if $\alpha : \mathbb{N} \to S_{\mathcal{BN}}$ is an infinite path.

A global state with no update steps is known as a *point attractor* and other complex, cyclic attractors can be considered (Schwab et al., 2020; Hopfensitz et al., 2013). Attractors provide important insights into a model's behaviour and can be associated with biological phenomena (Huang and Ingber, 2000).

## 2.2 A Compositional Framework

A range of approaches for composing and decomposing Boolean networks for analysis have been proposed (for example, see (Alkhudhayr and Steggles, 2019; Zhao et al., 2013; Mizera et al., 2017)). We focus on recent work which presents a novel approach to composing synchronous Boolean networks based on *merging entities between Boolean networks using Boolean operators* (Alkhudhayr and Steggles, 2019; Abdulrahman and Steggles, 2023). They focused solely on the synchronous update semantics and developed a range of interesting techniques and tools for analysing composed synchronous Boolean networks.

We briefly introduce key concepts from this compositional framework needed in the sequel. We only consider using conjunction for merging entities in the sequel but note that all the results presented also hold for disjunction. We begin by recalling the definition of a composition (Abdulrahman and Steggles, 2023).

**Definition 3.** *A composition $\Sigma = (M, E)$, where $M = \{\mathcal{BN}_1, \mathcal{BN}_2, ..., \mathcal{BN}_n\}$ is the set of Boolean networks (submodels) that are used in the composition $\Sigma$, for some $n \in \mathbb{N}$, $n > 1$, and $E$ defines merged entities*

$$E \subseteq \{\{g_1, g_2\} \mid \mathcal{BN}_i, \mathcal{BN}_j \in M,\ \mathcal{BN}_i \neq \mathcal{BN}_j \text{ and } g_1 \in G_i,\ g_2 \in G_j\}$$

As an illustrative example, consider the composition $\Sigma_{Ex} = (M_{Ex}, E_{Ex})$ presented in Figure 2, where $M_{Ex} = \{\mathcal{BN}_{Ex1}, \mathcal{BN}_{Ex2}, \mathcal{BN}_{Ex3}\}$, and $E_{Ex} = \{\{g_1^1, g_1^2\}, \{g_2^1, g_2^3\}, \{g_2^2, g_1^3\}\}$.

In order to reason about a composition we introduce the following important notations and definitions. We let $gc(\Sigma, \mathcal{BN}_i)$ represent the set of entities from $\mathcal{BN}_i$ that are composed in $\Sigma$ and define it as $gc(\Sigma, \mathcal{BN}_i) = \{g \mid g \in G_i$ *and* $\{g, g'\} \in E\}$. We let $gc(\Sigma) = gc(\Sigma, \mathcal{BN}_1) \cup \ldots \cup gc(\Sigma, \mathcal{BN}_n)$ represent the set of all entities that are composed in $\Sigma$.

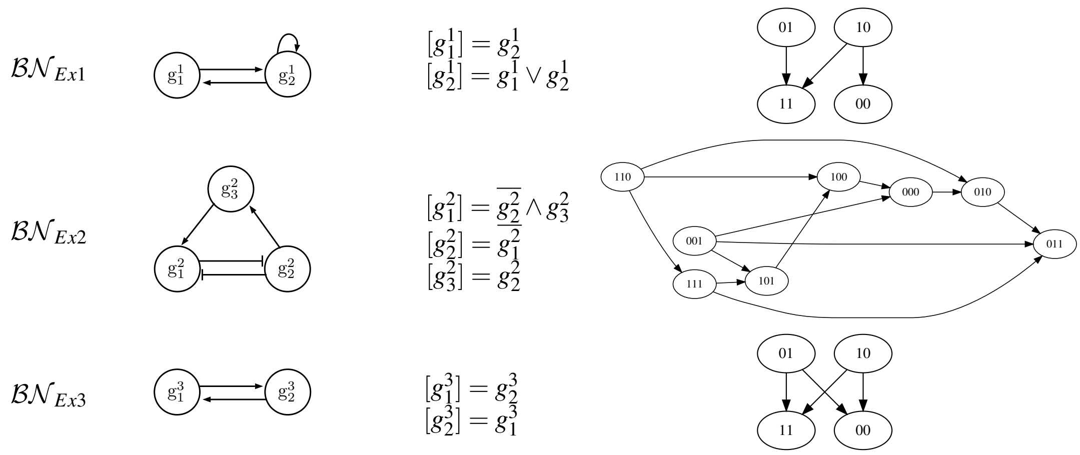


Figure 1: The interaction diagram, next-state functions and the state graph for three example Boolean networks $\mathcal{BN}_{Ex1}$, $\mathcal{BN}_{Ex2}$ and $\mathcal{BN}_{Ex3}$.

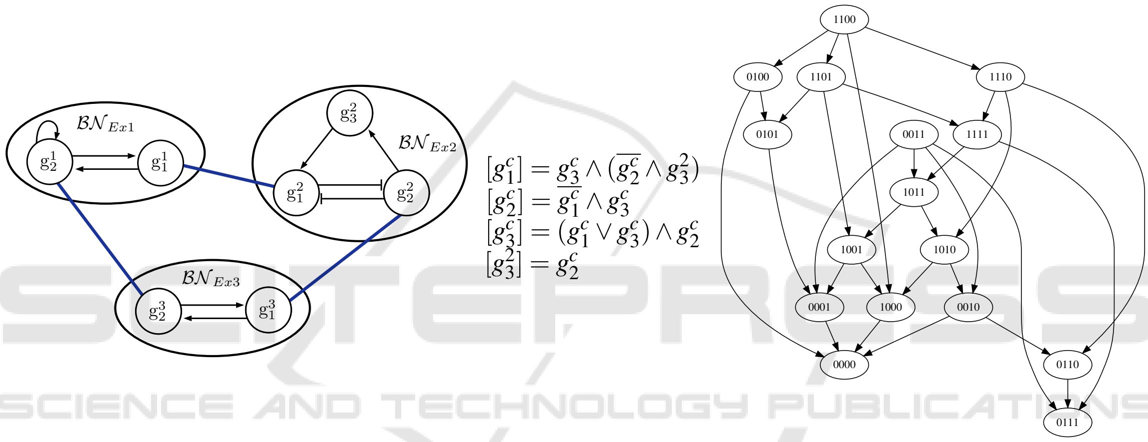


Figure 2: An example composition $\Sigma_{Ex}$ in which three Boolean networks $\mathcal{BN}_{Ex1}$, $\mathcal{BN}_{Ex2}$ and $\mathcal{BN}_{Ex3}$ are composed, where the resulting compositional entities (represented by thick blue edges) are $g_1^c = \{g_1^1, g_1^2\}$, $g_2^c = \{g_2^2, g_1^3\}$ and $g_3^c = \{g_2^1, g_2^3\}$. The asynchronous state graph is also depicted for the resulting Boolean network $\mathcal{BN}(\Sigma_{Ex})$.



For any $g \in gc(\Sigma)$ we let $\Delta(g)$ denote the set of all entities that are composed with $g$ (including $g$ itself).

We let $\lambda(g)$ denote the index of the Boolean network that $g$ belongs to. For any $i \in \{1,\dots,n\}$ and any $g \in G_i$ we define $\lambda(g) = i$.

We let $\Sigma(g)$ be the renaming used to move between a submodel and the composition defined for any entity $g \in (G_1 \cup \ldots \cup G_n)$ by $\Sigma(g) = g$, if $g \notin gc(\Sigma)$; or $\Sigma(g) = \Delta(g)$, otherwise.

A composition $\Sigma$ results in a Boolean network $\mathcal{BN}(\Sigma)$ (Abdulrahman and Steggles, 2023).

**Definition 4.** *Let $\Sigma = (M, E)$ be a composition. Then we define the Boolean network $\mathcal{BN}(\Sigma) = (G(\Sigma), N(\Sigma), F(\Sigma))$ that results from $\Sigma$ as follows.*
***1. Entities:*** *$G(\Sigma) = \Sigma(G_1 \cup \ldots \cup G_n)$.*
***2. Neighbourhood:*** *for any entity $h \in G(\Sigma)$, we define the neighbourhood $N(\Sigma)(h)$ by*

$$N(\Sigma)(h) = \bigcup_{g \in h} \Sigma(N_{\lambda(g)}(g)),$$

*if $h = \Delta(g')$, for some $g' \in gc(\Sigma)$; or $N(\Sigma)(h) = \Sigma(N_{\lambda(h)}(h))$, otherwise.*
***3. Functions:*** *For any entity $g \in G(\Sigma)$, we define the next–state function $F(\Sigma)(g)$ on any $S \in S_{\mathcal{BN}(\Sigma)}$ by*

$$F(\Sigma)(g)(S[N(\Sigma)(g)]) = \bigwedge_{h \in \Delta(g')} F_{\lambda(h)}(h)(S[\Sigma(N_{\lambda(h)}(h))]),$$

*if $g = \Delta(g')$, for some $g' \in gc(\Sigma)$; or*

$$F(\Sigma)(g)(S[N(\Sigma)(g)]) = F_{\lambda(g)}(g)(S[\Sigma(N_{\lambda(g)}(g))]),$$

*otherwise.*

For an example, see Figure 2 where the composed model $\mathcal{BN}(\Sigma_{Ex})$ resulting from $\Sigma_{Ex}$ is shown. In a slight abuse of notation we use the set of entities composed as the name of the resulting composed entity.

In the sequel we use the above definitions for our asynchronous compositional theory but place an important restriction on a composition's graph structure to ensure that no two entities in an asynchronous submodel are composed together (since this would result in synchronous behaviour).

### 2.3 Related Work

The compositional framework for asynchronous Boolean networks we present is based on the novel idea of composing synchronous Boolean networks by merging entities using a Boolean operator (Alkhudhayr and Steggles, 2019; Abdulrahman and Steggles, 2023). However, the work presented here takes a significantly different approach as required by our focus on asynchronous models.

There are some interesting studies on behaviour preservation and subnetwork embeddings in the literature (for example, see (Thomas and d'Ari, 1990)) but the approach we take is new.

A range of research into the composition and decomposition of Boolean networks can be found in the literature, e.g. (Tournier and Chaves, 2013; Dubrova and Teslenko, 2005; Cheng et al., 2012; Zhao et al., 2013). The approaches taken by these papers appear to be significantly different to our work and focus on applying decomposition/composition to aid identifying the attractors of a Boolean network. They appear not to consider developing a general compositional framework for constructing and analysing asynchronous Boolean networks as we do here.

## 3 BEHAVIOUR PRESERVATION

### 3.1 Asynchronous Interference

A key concept in the synchronous compositional framework (Alkhudhayr and Steggles, 2019) is the behavioural interference that can occur when entities are merged in a composition. To illustrate this idea, consider the situation where two entities $g_1$ and $g_2$ are merged using conjunction. Then interference would occur if $g_1$ wanted to transition to 1 but $g_2$ to 0 since the composed entity would transition to $1 \wedge 0 = 0$ (i.e. the behaviour of entity $g_1$ has been interfered with to become 0). To capture the impact interference can have on a submodel's behaviour the *interference state graph* was developed in (Alkhudhayr and Steggles, 2019) by extending the state graph for a Boolean network with edges that could result due to interference.

We formulate a new definition for an interference state graph for asynchronous compositions by extending the asynchronous state graph $SG(\mathcal{BN}_i)$ with two new types of edges to reflect the new asynchronous behaviour possible in a composition. Firstly, we add self-loops to every global state in the state graph $SG(\mathcal{BN}_i)$ to capture $\mathcal{BN}_i$ needing to remain at a state while an entity in another submodel of the composition updates. Secondly, we add interference edges to capture when the state of an entity in $\mathcal{BN}_i$ involved in a composition is 1 and so can be forced to update to 0 by interference from another submodel.

**Definition 5.** *Let $\mathcal{BN}_i = \{G_i, N_i, F_i\}$ and let $X \subseteq G_i$. Then an asynchronous interference state graph $AISG_X(\mathcal{BN}_i) = (S_{\mathcal{BN}_i}, \xrightarrow[X]{\mathcal{BN}_i})$, where $\xrightarrow[X]{\mathcal{BN}_i} = \xrightarrow{\mathcal{BN}_i} \cup\ \varepsilon \cup \kappa$, and $\varepsilon = \{(S,S) \mid S \in S_{\mathcal{BN}_i}\}$ and $\kappa = \{(S, S[h \to 0]) \mid h \in X,\ S \in S_{\mathcal{BN}_i},\ S(h) = 1\}$.*

For any $i \in \{1,...,n\}$, we let $I_i$ denote the asynchronous interference state graph $AISG_{gc(\Sigma,\mathcal{BN}_i)}(\mathcal{BN}_i)$ for $\mathcal{BN}_i$ over $\Sigma$. As an example, consider Figure 3 which presents the asynchronous interference state graph $I_{Ex1} = AISG_{\{g_1^1, g_2^1\}}(\mathcal{BN}_{Ex1})$ for $\mathcal{BN}_{Ex1}$ over $\Sigma_{Ex}$.

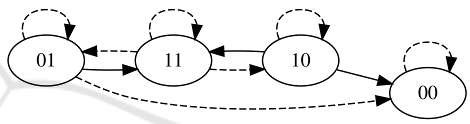


Figure 3: The asynchronous interference state graph of $\mathcal{BN}_{Ex1}$ in composition $\Sigma_{Ex}$, where dashed edges represent new transitions arising from interference.

The following theorem shows the asynchronous interference state graph bounds the behaviour a Boolean network can exhibit in a composition.

**Theorem 1.** *Let $\Sigma = (M, E)$, where $M = \{\mathcal{BN}_1, ..., \mathcal{BN}_n\}$. Then for any $i = \{1,...,n\}$, we have $Path(SG(\mathcal{BN}(\Sigma)))[\Sigma(G_i)] \subseteq Path(I_i)$.*

*Proof.* Let $i \in \{1,...,n\}$ and let path $\beta \in Path(SG(\mathcal{BN}(\Sigma)))$. Then we have to show that there is a path $\alpha \in Path(I_i)$ such that $\beta[\Sigma(G_i)] = \alpha$. It suffices to show that for any $k \in SD(\beta)$ there exists an edge

$$\beta(k)[\Sigma(G_i)] \xrightarrow{I_i} \beta(k+1)[\Sigma(G_i)] \quad (1)$$

in the interference state graph $I_i$. The transition (1) must have resulted from updating an entity $g \in G(\Sigma)$ and we therefor have three possible cases to consider based on entity $g$.

**Case 1:** Suppose $g \notin \Sigma(G_i)$ (i.e. the state update step involved an entity unassociated with $\mathcal{BN}_i$). It follows that $\beta(k)[\Sigma(G_i)] = \beta(k+1)[\Sigma(G_i)]$. By the definition of the asynchronous interference state graph (Definition 5), we know that $\beta(k)[\Sigma(G_i)] \xrightarrow{I_i} \beta(k)[\Sigma(G_i)]$ holds (self-loop) and so we must have (1) holds as required.

**Case 2:** Suppose $g \in (G_i \backslash gc(\Sigma, \mathcal{BN}_i))$ (i.e. the state update step involved an entity in $G_i$ that is not used in the composition). It must follow by the definition of $\mathcal{BN}(\Sigma)$ that $\beta(k)[\Sigma(G_i)] \xrightarrow{\mathcal{BN}_i} \beta(k+1)[\Sigma(G_i)]$ and so (1) must hold by the definition of asynchronous interference state graph.
**Case 3:** Suppose $g = \Delta(h)$, for some $h \in gc(\Sigma, \mathcal{BN}_i)$ (i.e. the state update step involved a composed entity associated with $\mathcal{BN}_i$). Then, there are two sub cases to consider based on if entity $h$ updates or not in $\mathcal{BN}_i$. The proof of these cases are based on the definition of the asynchronous interference state graph and are omitted here for brevity (see (Alshahrani, 2024) for details). □

## 3.2 Behaviour Preservation

Often when analysing a compositional model we want to check whether the behaviour of a submodel has been preserved. In this section we reformulate and extend important behavioural preservation results from (Alkhudhayr and Steggles, 2019) to asynchronous compositions. We begin by recalling the definition of *compatibility* (Alkhudhayr and Steggles, 2019) which formalises the idea of preserving a submodel's behaviour in a composition.

**Definition 6.** *Let $\Sigma = (M, E)$ be a composition with $M = \{\mathcal{BN}_1, ..., \mathcal{BN}_n\}$, and let $i \in \{1, ..., n\}$. Then, $\mathcal{BN}_i$ is said to be* compatible *with $\Sigma$ iff $Path(SG(\mathcal{BN}_i)) \subseteq Path(SG(\mathcal{BN}(\Sigma)))[\Sigma(G_i)]$.*

As an example, consider the composition $\Sigma_{Ex}$ (Figure 2); it can be shown that $\mathcal{BN}_{Ex2}$ is compatible with $\Sigma_{Ex}$ but that $\mathcal{BN}_{Ex1}$ and $\mathcal{BN}_{Ex3}$ are not.

In order to check compatibility compositionally a property called *weak alignment* was proposed in (Alkhudhayr and Steggles, 2019) which compositionally characterises compatibility using interference state graphs. We update and extend this approach to the asynchronous case. This provides important insights into the semantic differences that exist between the synchronous and asynchronous update semantics. We begin by recalling how submodel states and paths can be merged (Abdulrahman and Steggles, 2023).

**Definition 7.** *Let $S_i \in S_{\mathcal{BN}_i}$, for $i \in \{1, ..., n\}$. We let $\wedge_\Sigma(S_1, ..., S_n) \in S_{\mathcal{BN}(\Sigma)}$ be the global state that results from merging $S_1, \ldots, S_n$ defined for any $g \in G(\Sigma)$ by $\wedge_\Sigma(S_1, ..., S_n)(g) = \bigwedge_{h \in g} S_{\lambda(h)}(h)$, if $g = \Delta(g')$, for some $g' \in gc(\Sigma)$; or $\wedge_\Sigma(S_1, ..., S_n)(g) = S_{\lambda(g)}(g)$, otherwise. Let $\alpha_i \in Path(I_i)$, for $i \in \{1, ..., n\}$, such that $SD(\alpha_1) = \cdots = SD(\alpha_n)$. Then we let $\wedge_\Sigma(\alpha_1, ..., \alpha_n)$ represent the path resulting from merging paths $\alpha_1, \ldots, \alpha_n$ defined for any $m \in SD(\alpha_1)$ by $\wedge_\Sigma(\alpha_1, ..., \alpha_n)(m) = \wedge_\Sigma(\alpha_1(m), \ldots, \alpha_n(m))$.*

Next we introduce what it means for submodel paths to align such that they can be merged to form behaviour in the composed model.

**Definition 8.** *Let $\alpha_i \in Path(I_i)$, for $i \in \{1, ..., n\}$. We say $\alpha_1, \ldots, \alpha_n$ align (on $\Sigma$) iff we have*
*1) $SD(\alpha_1) = SD(\alpha_2) = \cdots = SD(\alpha_n)$; and*
*2) for every $g \in gc(\Sigma)$, every $h \in \Delta(g)$ and every $m \in SD(\alpha_1)$, we have $\alpha_{\lambda(g)}(m)(g) = \alpha_{\lambda(h)}(m)(h)$.*
*We say $\alpha_1, \ldots, \alpha_n$ update align (on $\Sigma$) iff they align and for each $m \in SD(\alpha_1)$ there exists $i \in \{1, ..., n\}$ and $g \in G_i$ such that $\alpha_i(m+1)(g) = \neg\alpha_i(m)(g)$ and $\alpha_i(m) \xrightarrow{\mathcal{BN}_i} \alpha_i(m+1)$, and for all $k \in \{1, ..., n\}$, $k \neq i$, and all $h \in G_k$ we have $\alpha_k(m)(h) = \alpha_k(m+1)(h)$, if $g \notin gc(\Sigma, \mathcal{BN}_i)$ or $h \notin \Delta(g)$.*

Update alignment is an important new property which captures when a collection of paths can be merged asynchronously to produce a well-defined composed path. The following are important results about merging aligned and update aligned paths.

**Lemma 1.** *Let $\alpha_1 \in Path(I_1), \ldots, \alpha_n \in Path(I_n)$, such that $\alpha_1, \ldots, \alpha_n$ align. Then, for every $k \in \{1, ..., n\}$, we have $\wedge_\Sigma(\alpha_1, ..., \alpha_n)[\Sigma(G_k)] = \alpha_k$.*

*Proof.* This is straightforward to prove (see (Alshahrani, 2024) for details). □

**Lemma 2.** *Let $\alpha_1 \in Path(I_1), \ldots, \alpha_n \in Path(I_n)$ be paths such that $\alpha_1, \ldots, \alpha_n$ update align. Then we have $\wedge_\Sigma(\alpha_1, ..., \alpha_n) \in Path(SG(\mathcal{BN}(\Sigma)))$.*

*Proof.* Let $\alpha_1 \in Path(I_1), \ldots, \alpha_n \in Path(I_n)$ such that $\alpha_1, \ldots, \alpha_n$ update align. To show that $\wedge_\Sigma(\alpha_1, ..., \alpha_n) \in Path(SG(\mathcal{BN}(\Sigma)))$ it suffices to show that for each $m \in SD(\alpha_1)$, there exists $g \in G(\Sigma)$ such that

$$\begin{aligned} &\wedge_\Sigma(\alpha_1(m+1), \ldots, \alpha_n(m+1))(g) = \\ &F(\Sigma)(g)(\wedge_\Sigma(\alpha_1(m), \ldots, \alpha_n(m))[N(\Sigma)(g)]) = \\ &\neg \wedge_\Sigma (\alpha_1(m), \ldots, \alpha_n(m))(g) \end{aligned} \quad (1)$$

and for all other $g' \in G(\Sigma)$, $g \neq g'$, we have

$$\begin{aligned} &\wedge_\Sigma(\alpha_1(m+1), \ldots, \alpha_n(m+1))(g') = \\ &\wedge_\Sigma(\alpha_1(m), \ldots, \alpha_n(m))(g') \end{aligned} \quad (2)$$

By the assumption of update alignment we know there exists $i \in \{1, ..., n\}$ and $h \in G_i$ such that $\alpha_i(m+1)(h) = \neg\alpha_i(m)(h)$. Then to show (1) and (2), there are two cases to consider based on whether entity $h$ is used in the composition or not. The proof of these cases is based on using the definition of $\mathcal{BN}(\Sigma)$ and Lemma 1, and are omitted for brevity (see (Alshahrani, 2024) for details). □

We can now formulate the asynchronous version of the weak alignment property.

**Definition 9.** *For any $i \in \{1,\ldots,n\}$, we say $\mathcal{BN}_i$ is update weakly aligned (on $\Sigma$) iff for every $\alpha_i \in Path(SG(\mathcal{BN}_i))$, there exists $\alpha_k \in Path(I_k)$, for each $k \in \{1,...,n\}$, $k \neq i$, such that $\alpha_1,\ldots,\alpha_n$ update align.*

We now prove that update weak alignment compositionally characterises compatibility.

**Theorem 2.** *For $i \in \{1,...,n\}$, we have $\mathcal{BN}_i$ is compatible on $\Sigma$ iff $\mathcal{BN}_i$ is update weakly aligned on $\Sigma$.*

*Proof.* Let $i \in \{1,...,n\}$.
1) Suppose $\mathcal{BN}_i$ is compatible on $\Sigma$. We need to show that $\mathcal{BN}_i$ is update weakly aligned on $\Sigma$. Let $\alpha_i \in Path(SG(\mathcal{BN}_i))$. Then by assumption of compatibility there must exist a path $\beta \in Path(SG(\mathcal{BN}(\Sigma)))$ such that $\alpha_i = \beta[\Sigma(G_i)]$. For each $k \in \{1,...,n\}$, $k \neq i$, let $\alpha_k = \beta[\Sigma(G_k)]$. Then, by Theorem 1 we know that $\alpha_k \in Path(I_k)$, for each $k \in \{1,...,n\}, k \neq i$. It remains to show that $\alpha_1,\ldots,\alpha_n$ update align. Since $\alpha_1,\ldots,\alpha_n$ are all projected from the same path $\beta$ it follows that $\alpha_1,\ldots,\alpha_n$ must align. Furthermore, as $\alpha_i \in Path(SG(\mathcal{BN}_i))$ we know that for every $m \in SD(\alpha_1)$ there must exist an entity $h \in G_i$ which updates such that $\alpha_i(m+1)(h) = \neg\alpha_i(m)(h)$. There are now two cases to consider based on if entity $h$ is used in the composition or not. These cases are proved using the definition of $\mathcal{BN}(\Sigma)$ and the definition of an asynchronous update (Definition 2). We omit them for brevity (see (Alshahrani, 2024) for details).
2) Suppose $\mathcal{BN}_i$ is weakly aligned on $\Sigma$. Then we need to show that $\mathcal{BN}_i$ is compatible on $\Sigma$, that is $Path(SG(\mathcal{BN}_i)) \subseteq Path(SG(\mathcal{BN}(\Sigma)))[\Sigma(G_i)]$. To prove this, consider any path $\alpha_i \in Path(SG(\mathcal{BN}_i))$. Then, by update weak alignment we know there must exist paths $\alpha_k \in Path(I_k)$, for $k \in \{1,...,n\}$, $k \neq i$, such that $\alpha_1,\ldots,\alpha_n$ update align. It follows by Lemma 1 that $\wedge_\Sigma(\alpha_1,...,\alpha_n)[\Sigma(G_i)] = \alpha_i$ and by Lemma 2 that $\wedge_\Sigma(\alpha_1,...,\alpha_n) \in Path(SG(\mathcal{BN}(\Sigma)))$ as required. $\square$

## 3.3 Behaviour Preservation Based on Asynchronous Sequences

The concept of compatibility introduced previously was based on ideas formulated for synchronous models (Alkhudhayr and Steggles, 2019). In the asynchronous case it is possible for a submodel to pause it's behaviour while another submodel updates and this can result in a submodel's projected path in a composition containing repeated states. To take account of this we consider reducing paths by removing any duplicated consecutive states. We formalise this using $red(\alpha)$ which takes a path $\alpha \in Path(SG(\mathcal{BN}(\Sigma)))[\Sigma(G_i)]$ and removes any duplicated consecutive states. These ideas lead to a new notion of compatibility called *sequence compatible*.

**Definition 10.** *Let $\Sigma = (M,E)$ be a composition with $M = \{\mathcal{BN}_1,...,\mathcal{BN}_n\}$, and let $i \in \{1,...,n\}$. Then, $\mathcal{BN}_i$ is sequence compatible (on $\Sigma$) iff*

$$Path(SG(\mathcal{BN}_i)) \subseteq red(Path(SG(\mathcal{BN}(\Sigma)))[\Sigma(G_i)])$$

It can be seen that $\mathcal{BN}_{Ex3}$ is sequence compatible on $\Sigma_{Ex}$ even though it was not compatible. We can also see that $\mathcal{BN}_{Ex2}$ is both compatible and sequence compatible. Note it can be shown that compatibility implies sequence compatibility.

In order to compositionally characterise sequence compatibility we adapt the definition of update weak alignment to take account of reduced paths. For any $\alpha \in Path(SG(\mathcal{BN}_i))$, we let $Exp(\alpha) = \{\alpha' \mid \alpha' \in Path(I_i),\ red(\alpha') = \alpha\}$.

**Definition 11.** *For any $i \in \{1,...,n\}$, we say $\mathcal{BN}_i$ is sequence weakly aligned (on $\Sigma$) iff for every path $\alpha \in Path(SG(\mathcal{BN}_i))$, there exists a path $\alpha_i \in Exp(\alpha)$, and paths $\alpha_k \in Path(I_k)$, for each $k \in \{1,...,n\}$, $k \neq i$, such that $\alpha_1,\ldots,\alpha_n$ update align.*

We now prove that sequence weak alignment compositionally characterises sequence compatibility.

**Theorem 3.** *For any $i \in \{1,...,n\}$, we have $\mathcal{BN}_i$ is sequence compatible on $\Sigma$ iff $\mathcal{BN}_i$ is sequence weakly aligned on $\Sigma$.*

*Proof.* Let $i \in \{1,...,n\}$.
1) Suppose $\mathcal{BN}_i$ is sequence compatible on $\Sigma$. Then we need to show $\mathcal{BN}_i$ is sequence weakly aligned on $\Sigma$. Suppose $\alpha \in Path(SG(\mathcal{BN}_i))$. By sequence compatibility we know there exists $\beta \in Path(SG(\mathcal{BN}(\Sigma)))$ such that $red(\beta[\Sigma(G_i)]) = \alpha$. Let $\alpha_k = \beta[\Sigma(G_k)]$, for each $k \in \{1,...,n\}$. By Theorem 1 we know $\alpha_k \in Path(I_k)$, for each $k \in \{1,...,n\}$, and since $red(\beta[\Sigma(G_i)]) = \alpha$ we have $\alpha_i \in Exp(\alpha)$. It remains to show $\alpha_1,\ldots,\alpha_n$ update align. Since $\alpha_1,\ldots,\alpha_n$ are projected from $\beta$ it follows that they align. Furthermore, since $\beta \in Path(SG(\mathcal{BN}(\Sigma)))$ we know that for any $m \in SD(\beta)$ there must exist $g \in G(\Sigma)$ such that $\beta(m+1)(g) = \neg\beta(m)(g)$ and for all $h \in G(\Sigma)$, $h \neq g$, we have $\beta(m+1)(h) = \beta(m)(h)$. Given above it is straightforward to show that update alignment holds using two cases based on whether or not $g$ is a composed entity.
2) Suppose $\mathcal{BN}_i$ is sequence weakly aligned on $\Sigma$. Then we need to show $\mathcal{BN}_i$ is sequence compatible on $\Sigma$. Consider any path $\alpha \in Path(SG(\mathcal{BN}_i))$. Then by assumption of sequence weak alignment, there exists $\alpha_i \in Exp(\alpha)$, and $\alpha_k \in Path(I_k)$, for $k \in \{1,...,n\}$, $k \neq i$, such that $\alpha_1,\ldots,\alpha_n$ update align. It follows by Lemma 2 that

$\wedge_\Sigma(\alpha_1,...,\alpha_n) \in Path(SG(\mathcal{BN}(\Sigma)))$. Furthermore, by Lemma 1, we have $\wedge_\Sigma(\alpha_1,...,\alpha_n)[\Sigma(G_i)] = \alpha_i$, and so by assumption $\alpha_i \in Exp(\alpha)$, we have $\alpha = red(\wedge_\Sigma(\alpha_1,...,\alpha_n)[\Sigma(G_i)])$ holds as required. $\square$

# 4 POINT ATTRACTORS

Boolean networks can exhibit cyclic behaviour known as *attractors* (Hopfensitz et al., 2013) and identifying attractors is a crucial analysis step as they provide important practical insights into a Boolean network's behaviour (for example, see (Huang and Ingber, 2000; Saadatpour et al., 2011)).

In this section we develop an important new compositional technique for identifying *point attractors* (global states with no update steps) in an asynchronous composition. It is based on identifying potential self-loop states in submodels and then merging these when they align to construct point attractors. The approach is summarised below.

**1) Identify Potential Self-Loop States in Submodels.** We identify states in each submodel which can remain constant either because they are point attractors or due interference. We refer to these as *self-loop states* and for $i \in \{1,...,n\}$, define the set of self-loop states

$$SL(\mathcal{BN}_i) = \{S \mid S \in S_{\mathcal{BN}_i},\ \omega(S)\},$$

where $\omega(S)$ is true iff for every $S' \in U_{\mathcal{BN}_i}(S)$, exists $h \in gc(\Sigma, \mathcal{BN}_i)$ such that $S(h) = 0$, and $S'(h) = 1$.

**2) Identify Aligned State Tuples.** Next we group self-loop states that align to form state tuples. Let $S_i \in SL(\mathcal{BN}_i)$, for $i \in \{1,...,n\}$. Then $(S_1,...,S_n)$ is an *aligned state tuple* (over $\Sigma$) iff for every $g \in gc(\Sigma)$, and for all $h \in \Delta(g)$, we have $S_{\lambda(g)}(g) = S_{\lambda(h)}(h)$. We denote the set of all aligned states tuples over a composition $\Sigma$ as $alignST(\Sigma)$.

**3) Merge Valid Aligned State Tuples.** We say $(S_1,...,S_n) \in alignST(\Sigma)$ is *valid* iff for all $g \in gc(\Sigma)$, if $S_{\lambda(g)}(g) = 0$, then there exists $h \in \Delta(g)$ such that $F_{\lambda(h)}(h)(S_{\lambda(h)}[N_{\lambda(h)}(h)]) = 0$. Each valid aligned state tuple $(S_1,...,S_n) \in alignST(\Sigma)$ represents a point attractor $\wedge_\Sigma(S_1,...,S_n)$ in $\mathcal{BN}(\Sigma)$.

To illustrate the approach consider compositionally identifying the point attractors in $\mathcal{BN}(\Sigma_{Ex})$ (Figure 2). We first identify the self-loop set for each submodel: $SL(\mathcal{BN}_{Ex1}) = \{11,00,01\}$; $SL(\mathcal{BN}_{Ex2}) = \{000,011\}$; and $SL(\mathcal{BN}_{Ex3}) = \{11,00\}$. Next, we generate the aligned state tuples $alignST(\Sigma_{Ex}) = \{(00,000,00),(01,011,11)\}$. Both of these aligned state tuples are valid and so are merged to form point attractors: $\wedge_{\Sigma_{Ex}}(00,000,00) = 0000$ and $\wedge_{\Sigma_{Ex}}(01,011,11) = 0111$.

It remains to show that our approach is correct by proving it is *sound* and *complete*.

**Theorem 4.** *(Soundness) Let $(S_1,...,S_n) \in alignST(\Sigma)$ be a valid aligned state tuple. Then $\wedge_\Sigma(S_1,...,S_n)$ is a point attractor in $\mathcal{BN}(\Sigma)$.*

*Proof.* Suppose $(S_1,...,S_n) \in alignST(\Sigma)$. Then we need to show that $\wedge_\Sigma(S_1,...,S_n)$ is a point attractor in $\mathcal{BN}(\Sigma)$. By definition of merging states (Definition 7), we know $\wedge_\Sigma(S_1,...,S_n) \in S_{\mathcal{BN}(\Sigma)}$. To prove that $\wedge_\Sigma(S_1,...,S_n)$ is a point attractor in $\mathcal{BN}(\Sigma)$, we need to show $U_{\mathcal{BN}(\Sigma)}(\wedge_\Sigma(S_1,...,S_n)) = \{\}$. To do this we show that for every $g \in G(\Sigma)$, we have

$$F(\Sigma)(g)(\wedge_\Sigma(S_1,...,S_n)[N(\Sigma)(g)]) = \wedge_\Sigma(S_1,...,S_n)(g) \tag{1}$$

There are two cases to consider based on whether entity $g$ is a composed entity or not.

**Case 1:** Suppose $g \in ((G_1 \cup ... \cup G_n) \setminus gc(\Sigma))$ (i.e. $g$ is not a composed entity). By the assumptions, we know $S_{\lambda(g)} \in SL(\mathcal{BN}_{\lambda(g)})$. Then the result follows by the alignment assumption, Lemma 1 and by the definition of $SL(\mathcal{BN}_{\lambda(g)})$.

**Case 2:** Suppose $g = \Delta(g')$, for some $g' \in gc(\Sigma)$ (i.e. $g$ is a composed entity). By definition of $\mathcal{BN}(\Sigma)$ and case assumption, we have $S_{\lambda(h)} \in SL(\mathcal{BN}_{\lambda(h)})$, for every $h \in g$. Then we have two subcases to consider:

**Case 2.1:** Suppose $\wedge_\Sigma(S_1,...,S_n)(g) = 0$. Then by assumptions, Lemma 1 and conjunction we have

$$\bigwedge_{h\in g} F_{\lambda(h)}(h)(\wedge_\Sigma(S_1,...,S_n)[\Sigma(N_{\lambda(h)}(h))]) = 0,$$

and so result follows by the definition of $\mathcal{BN}(\Sigma)$.

**Case 2.2:** Suppose $\wedge_\Sigma(S_1,...,S_n)(g) = 1$. Then by assumptions, Lemma 1 and conjunction we have

$$\bigwedge_{h\in g} F_{\lambda(h)}(h)(\wedge_\Sigma(S_1,...,S_n)[\Sigma(N_{\lambda(h)}(h))]) = 1$$

and so result follows by definition of $\mathcal{BN}(\Sigma)$. $\square$

**Theorem 5.** *(Completeness) Let $S \in S_{\mathcal{BN}(\Sigma)}$ be a point attractor in the composed model $\mathcal{BN}(\Sigma)$. Then, there must exist a valid aligned state tuple $(S_1,...,S_n) \in alignST(\Sigma)$ such that $\wedge_\Sigma(S_1,...,S_n) = S$.*

*Proof.* Let $S \in S_{\mathcal{BN}(\Sigma)}$ be a point attractor in $\mathcal{BN}(\Sigma)$. Then $U_{\mathcal{BN}(\Sigma)}(S) = \{\}$ and $F(\Sigma)(g)(S[N(\Sigma)(g)]) = S(g)$, for every $g \in G(\Sigma)$. Let $S_i = S[\Sigma(G_i)]$, for each $i \in \{1,...,n\}$. We know $S_1,...,S_n$ must align and so $(S_1,...,S_n) \in alignST(\Sigma)$. Furthermore, by Lemma 1 $\wedge_\Sigma(S_1,...,S_n) = S$. We have two properties to show:
i) Let $i \in \{1,...,n\}$. We need to show that $S_i \in SL(\mathcal{BN}_i)$. If $U_{\mathcal{BN}_i}(S_i) = \{\}$ then by definition $S_i \in SL(\mathcal{BN}_i)$. Alternatively, suppose $U_{\mathcal{BN}_i}(S_i) \neq \{\}$ and let $S'_i \in U_{\mathcal{BN}_i}(S_i)$. By the definition of conjunction and $\mathcal{BN}(\Sigma)$ it can be seen that interference must occur here to produce a self-loop and so $S_i \in SL(\mathcal{BN}_i)$.

ii) We must show that $(S_1,...,S_n)$ is valid. Suppose $g = \Delta(g')$, for some $g' \in gc(\Sigma)$, and $S(g) = 0$. Furthermore, suppose for a contradiction there is no required interference. Then by definition of $\mathcal{BN}(\Sigma)$ and conjunction we can contradict $S$ being a point attractor. It follows that $(S_1,...,S_n)$ must be valid. $\square$

## 5 CONCLUSIONS

In this paper we developed a range of new compositional techniques for constructing and analysing *asynchronous* Boolean networks by building on recent compositional work on synchronous Boolean networks (Alkhudhayr and Steggles, 2019; Abdulrahman and Steggles, 2023). The compositional framework developed provides a foundation for helping to address the current practical limitation of applying asynchronous Boolean networks and provides interesting insight into the differences between synchronous and asynchronous updating.

The key contributions of the paper are:

i) Formulated a new asynchronous version of the *interference state graph*, a key concept in the compositional framework (Alkhudhayr and Steggles, 2019; Abdulrahman and Steggles, 2023) and proved it bounds a submodel's compositional behaviour.

ii) Developed range of new compositional behaviour preservation results.

iii) Developed a new compositional technique for identifying point attractors which we formally showed to be correct.

We are now working to develop compositional techniques for identifying more general types of asynchronous cyclic and complex attractors (Hopfensitz et al., 2013; Schwab et al., 2020). The aim is to then develop tool support for compositionally analysing asynchronous Boolean networks and undertake large realistic case studies. We also intend to consider developing decompositional techniques.

We would like to thank Hanin Abdulrahman and the anonymous referees for their comments. We gratefully acknowledge the support provided by Faculty of Computer Science, King Khalid University.

## REFERENCES

Abdulrahman, H. and Steggles, J. (2023). *Compositional Techniques for Boolean Networks and Attractor Analysis*. LNCS 14150, Springer.

Alkhudhayr, H. and Steggles, J. (2019). A compositional framework for boolean networks. *Biosystems*, 186:103960.

Alshahrani, M. (2024). *Compositional Techniques and Tools for Asynchronous Boolean Networks*. PhD thesis, Newcastle University (to appear).

Barbuti, R., Gori, R., Milazzo, P., and Nasti, L. (2020). A survey of gene regulatory networks modelling methods: from differential equations, to boolean and qualitative bioinspired models. *Journal of Membrane Computing*, 2:207–226.

Cheng, D., Zhao, Y., Kim, J., and Zhao, Y. (2012). Approximation of boolean networks. In *Proceedings of the 10th World Congress on Intelligent Control and Automation*, pages 2280–2285.

Dahlhaus, M., Burkovski, A., Hertwig, F., Mussel, C., Volland, R., Fischer, M., Debatin, K.-M., Kestler, H. A., and Beltinger, C. (2016). Boolean modeling identifies greatwall/mastl as an important regulator in the aurka network of neuroblastoma. *Cancer Letters*, 371(1):79–89.

Dubrova, E. and Teslenko, M. (2005). Compositional properties of random boolean networks. *Physical Review E*, 71(5):056116.

Groote, J. F., Kouters, T. W., and Osaiweran, A. (2015). Specification guidelines to avoid the state space explosion problem. *Software Testing, Verification and Reliability*, 25(1):4–33.

Hopfensitz, M., Müssel, C., Maucher, M., and Kestler, H. A. (2013). Attractors in boolean networks: a tutorial. *Computational Statistics*, 28(1):19–36.

Huang, S. and Ingber, D. E. (2000). Shape-dependent control of cell growth, differentiation, and apoptosis: switching between attractors in cell regulatory networks. *Experimental cell research*, 261(1):91–103.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467.

Kauffman, S. A. et al. (1993). *The origins of order: Self-organization and selection in evolution*. Oxford University Press, USA.

Mizera, A., Pang, J., Qu, H., and Yuan, Q. (2017). A new decomposition method for attractor detection in large synchronous boolean networks. In *Int. Symp. on Dependable Software Engineering: Theories, Tools, and Applications*, pages 232–249. Springer.

Pandey, S., Wang, R.-S., Wilson, L., Li, S., Zhao, Z., Gookin, T. E., Assmann, S. M., and Albert, R. (2010). Boolean modeling of transcriptome data reveals novel modes of heterotrimeric g-protein action. *Molecular systems biology*, 6(1):372.

Saadatpour, A., Wang, R.-S., Liao, A., Liu, X., Loughran, T. P., Albert, I., and Albert, R. (2011). Dynamical and structural analysis of a t cell survival network identifies novel candidate therapeutic targets for large granular lymphocyte leukemia. *PLoS computational biology*, 7(11):e1002267.

Schwab, J. D., Kühlwein, S. D., Ikonomi, N., Kühl, M., and Kestler, H. A. (2020). Concepts in boolean network modeling: What do they all mean? *Computational and structural biotechnology journal*, 18:571–582.

Thomas, R. and d'Ari, R. (1990). *Biological feedback*. CRC press.

Tournier, L. and Chaves, M. (2013). Interconnection of asynchronous boolean networks, asymptotic and transient dynamics. *Automatica*, 49(4):884–893.

Zhao, Y., Kim, J., and Filippone, M. (2013). Aggregation algorithm towards large-scale boolean network analysis. *IEEE Transactions on Automatic Control*, 58(8):1976–1985.