Freie Universität Berlin

Fachbereich Mathematik und Informatik

Dissertation
zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.)

Combining Boolean Networks and Ordinary Differential Equations for Analysis and Comparison of Gene Regulatory Networks

Robert Schwieger

July 24, 2019

Supervisors
Prof. Dr. Heike Siebert
Prof. Dr. Susanna Röblitz
Examiners
Prof. Dr. Heike Siebert
Prof. Dr. Susanna Röblitz
Dr. habil. Élisabeth Remy

Date of the defense: 14.6.2019

Danksagung

Ohne den Autor, aber auch nicht ohne einige andere Menschen, wäre diese Arbeit nicht geschrieben worden. Ich möchte mich bei meinen Eltern bedanken, die mir den Rückhalt und das nötige Selbstvertrauen gegeben haben. Ich möchte mich bei meiner Familie – insbesondere meinem Onkel – bedanken, die mir von Kind an die Neugier und die Freude an der Wissenschaft vermittelt hat, und meinem Bruder, der diese Begeisterung mit mir teilt.

Mein Dank gilt ebenfalls meinen beiden Betreuerinnen Susanna und Heike, die neben der fachlichen Unterstützung, ein Klima geschaffen haben, in dem das Schreiben dieser Arbeit und die zugrunde liegende Forschung stets Freude gemacht haben. Ich möchte mich bei ihnen auch für das interessante Thema, die zahlreichen Anregungen und Ratschläge bedanken.

Special thanks to Elisa Tonello for her careful proofreading and valuable discussions. Special thanks to Markus Köbis for his help. Special thanks to Shane Carlson for his help with the English language. Thanks to my colleagues and former colleagues, especially Hannes Klarner, Adam Streck, Markus Köbis, Mohamed Omari and Melania Nowicka who shared with me the office, for stimulating discussions, a good time and helpful advices. Thanks also to Pedro T. Monteiro and Claudine Chaouiya for a nice and intesting stay at INESC-ID in Lisbon. Thanks to Danijela Žanko for her support. Thanks to my friends here in Berlin for distracting me occasionally from science and for their support and personal advice.

Contents

Nomenclature v

Chapter 0. Introduction 1
- 0.1. Positioning and motivation 1
- 0.2. Contributions of this thesis 3
- 0.3. Structure of this thesis 5

Chapter 1. Boolean models and ODE-models of gene regulatory networks 7
- 1.1. General notations 8
- 1.2. ODE models of regulatory networks 13
- 1.3. Boolean networks 16

Chapter 2. Conversion of Boolean models into ODE models 23
- 2.1. The transformation algorithm explained 24
- 2.2. Preservation of steady states and trap spaces 31
- 2.3. Application: a small example 35
- 2.4. Correspondence of trap spaces in detail 36
- 2.5. Discussion 44

Chapter 3. Operations on Boolean networks 46
- 3.1. Set operations on ASTGs 46
- 3.2. Quotient graphs of Boolean networks 49
- 3.3. Inverting transitions 60
- 3.4. Discussion 60

Chapter 4. Model pools and their graph representation 62
- 4.1. The qualitative state transition graph of a continuous monotonic model pool 64
- 4.2. The qualitative state transition graph of a Boolean monotonic model pool 75
- 4.3. The skeleton of the qualitative state transition graphs 86
- 4.4. Reconstruction of the interaction graph 93
- 4.5. Discussion 98

Chapter 5. Methods for applications 101
- 5.1. Distinguishing Boolean models with criteria based on states or transitions 102
- 5.2. Distinguishing Boolean models with criteria based on walks 121
- 5.3. Discussion 130

Chapter 6. Conclusion 132

Bibliography 134

Index 143

Appendix 145
Zusammenfassung 145
Declaration 147

Nomenclature

$(p, *)$ subspace or trapspace (Definition 1.27)

$[n]$ The set $\{1, \ldots, n\}$ (Section 1.1)

$G_{\text{async}}(f^{\Sigma})$ skeleton of Σ. (Definition 4.9)

$G_{\text{async}}(f) = (V_{\text{async}}(f), E_{\text{async}}(f))$ asynchronous state transition graph (Definition 1.23)

$\text{Auto}(f, I)$ Automaton accepting projected paths

$\mathbb{B}(n, m)$ the set of Boolean functions from $\{0,1\}^n$ to $\{0,1\}^m$. (Section 1.1.2)

$\overline{\mathscr{M}_{\mathbb{B}}}(\Sigma)$ Boolean Monotonic Model Pool (Definition 4.7)

cond^f The conditions for transitions corresponding to the Boolean function f (Definition 3.2)

$\deg^+(v)$ outdegree of the node v. (Definition 1.3)

$\deg^-(v)$ indegree of the node v. (Definition 1.3)

$\dot{x}$ derivative of x (Section 1.1.3)

$G^{\mathbb{B}}_{\text{QSTG}}(\Sigma) = \left(V^{\mathbb{B}}_{\text{QSTG}}(\Sigma), E^{\mathbb{B}}_{\text{QSTG}}(\Sigma)\right)$ Boolean QDE Graph (Definition 4.8)

$IG^{\text{global}}(f)$ global interaction graph of f (Definition 1.28)

$IG_f(x)$ local interaction graph of f (Definition 1.28)

ind^A indicator function of the set A (Definition 5.1)

$\leftrightarrow$ logical equivalence (Definition 1.12)

$\mathscr{M}(\Sigma, X)$ continuous monotonic model pool (Definition 4.3)

$\mathbb{F}_2$ the Galois field of two elements (Section 1.1.2)

$\mathbb{N}$ natural numbers not including zero (Section 1.1)

$\mathscr{P}(A)$ the powerset of A (Definition 1.1)

$G_{\text{QSTG}}(\Sigma) = (V_{\text{QSTG}}(\Sigma), E_{\text{QSTG}}(\Sigma))$ directed state-transition graph of the continuous monotonic model pool (Definition 4.5)

$\mid A \mid$ cardinality of the set A (Definition 1.1)

∇f gradient of f (Section 1.1.3)

$\neg$ logical negation (Definition 1.12)

$\oplus$ xor (Definition 1.12)

$\partial_j f$ partial derivative (Section 1.1.3)

proj^I projection on set of components $I \subseteq [n]$ (Definition 1.26)

$\text{diff}(E)$ All components that change on some edge in E. (Definition 1.2)

$\text{diff}(e)$ The index of the component which changes on the edge e. (Definition 1.2)

$\rightarrow$ logical implication (Definition 1.12)

Σ Sign matrix (Definition 4.7)

$\text{sign}(x)$ sign of x (Definition 4.2)

$\text{comm}(a, b)$ The indices of the components which are equal. (Definition 1.2)

$\text{Ess}(\varphi)$ Essential components of φ (Definition 5.2)

id identity function $\text{id} : \{0,1\}^n \rightarrow \{0,1\}^n$ (Proposition 3.1)

$\mathrm{pred}(v)$ predecessors of the node v. (Definition 1.3)

$\mathrm{StrongBasin}(A)$ Strong basin of attraction of an attractor A (Definition 1.10)

$\mathrm{succ}(v)$ sucessors of the node v. (Definition 1.3)

$\mathrm{WeakBasin}(A)$ Weak basin of attraction of an attractor A (Definition 1.10)

$\tilde{S}_{\mathscr{M}(\Sigma)}$ solution set (Definition 4.4)

$\vee$ logical or (Definition 1.12)

$\mathcal{I}(B)$ vanishing ideal of the set B (Section 5.1.4)

$\mathcal{V}(\mathcal{I})$ vanishing set of an ideal $\mathcal{I}$ (Section 5.1.4)

$\wedge$ logical and (Definition 1.12)

$\mathrm{diff}(a,b)$ The indices of the components which are different. (Definition 1.2)

$a \oplus^{-1} b$ $= a \leftrightarrow b$. (Definition 1.15)

$a \oplus^{0} b$ $= 0$. (Definition 1.15)

$a \oplus^{1} b$ $= a \oplus b$. (Definition 1.15)

$C(X,Y)$ Space of continuous functions from X to Y (Section 1.1.3)

$C^k(X,Y)$ Space of functions $f : X \to Y$ which have continuous derivatives $f^{(1)}, \dots, f^{(k)}$. (Section 1.1.3)

$E(G)$ set of edges of the graph G (Section 1.1.1)

e^A The transition e inverted on the set A (Definition 1.25)

f^Σ (asynchronous) update function of the skeleton of Σ. (Definition 4.9)

$G/\sim$ quotient graph derived with the equivalence relation $\sim$ (Definition 3.3)

G/φ quotient graph induced by the function φ (Definition 3.4)

$G_\prec(\mathcal{I})$ reduced Gröbner basis of an ideal $\mathcal{I}$ with respect to $\prec$ (Definition 5.5)

$in_\prec(\mathcal{I})$ initial ideal (Definition 5.5)

$in_\prec(p)$ initial monomial of a polynomial p (Definition 5.5)

J_f Jacobian matrix of f (Section 1.1.3)

$LT(p)$ leading term of a polynomial p (Section 5.1.4)

$S_{\mathscr{M}(\Sigma,X)}$ solution set (Definition 4.3)

$v \rightsquigarrow w$ there is a directed path from v to w (Definition 1.5)

$V(G)$ set of nodes of the graph G (Section 1.1.1)

v^A The node v inverted on the set A (Definition 1.24)

CHAPTER 0

Introduction

This thesis is concerned with different classes of models of gene regulatory networks. In this thesis we will learn more about the relations between these modeling frameworks and how models can be analyzed. Since the focus lies on the mathematical analysis of these models, only a rudimentary biological understanding is necessary. Nevertheless, we will begin with a very brief biological background that serves as a motivation for the mathematical concepts introduced later.

0.1. Positioning and motivation

What is a gene regulatory network? A *gene (or genetic) regulatory network* (GRN) is a network whose components represent molecular regulators in the cell governing gene expression levels of *Messenger RNA* (mRNA) and protein concentrations. The connections in a GRN represent the interactions of these molecular regulators. The molecular regulators can be proteins, RNA, DNA, or complexes of these. In other words, a gene regulatory network describes the interactions between genes and their products.

What role do GRNs play in organisms? Within single cells, regulatory networks constitute a way to process information. They can be thought of as "computational devices of living cells" Barkai and Leibler [1997]. For example, a yeast cell in a sugar solution will turn on genes that make enzymes for converting the sugar into alcohol, which are otherwise not produced Lee et al. [2002]. This effect is called *enzymatic adaption* Jacob and Monod [1961]. In other words, regulatory networks can react to the external environment of a cell.

Such a functionality is often realized by proteins acting as *transcription factors* binding to the *regulatory sites* of other genes De Jong [2004]. In a GRN, some of these transcription factors function for the cell as sensors of stimuli such as temperature, osmotic pressure, biological signaling molecules from other cells, nutrients, or harmful chemicals. The effect of their binding can be either activating or inhibiting. A representation of such a network is depicted in Figure 0.1.1.

Since transcription factors are encoded by genes themselves, which are possibly affected by other transcription factors (and so on) the picture becomes in general a much more complicated than the one depicted in Figure 0.1.1.

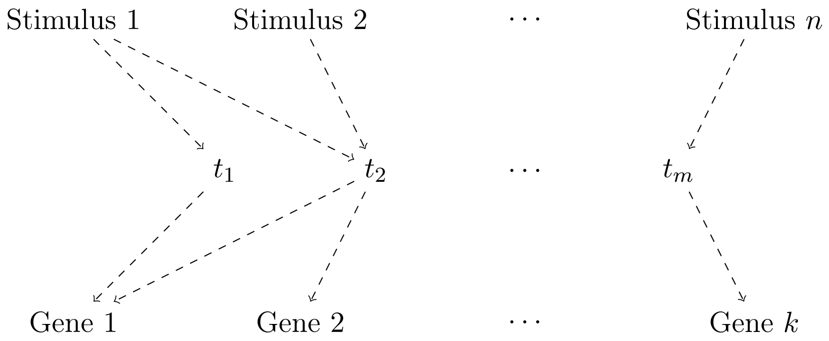


FIGURE 0.1.1. Environmental stimuli (first layer) are encoded by the transcription rates of transcription factors (second layer), which influence the transcription rate of genes (third layer) (see also [Alon, 2007, p.7]).

The role of GRNs is not restricted to cellular information processing. GRNs are also closely related to *epigenetics*. All cells in an organism contain the same genetic material[1], but their metabolism can differ considerably. Differences in the "programs" of gene expression that can be found in different cell types in multi-cellular organisms almost always occur without differences in the DNA sequence Reik [2007]. By definition, cell differentiation is therefore an example of epigenetics, which studies heritable changes in the phenotype that do not involve alterations in the DNA sequence Dupont et al. [2009]. Such heritable changes in the phenotype are based on GRNs, but act on a much slower time-scale.

How can we understand GRN function? In order to understand the functionality of GRNs, suitable mathematical formalisms are necessary for understanding the complex interactions among the components of GRNs. Such formalisms make it possible to simulate GRNs and analyze the fundamental properties of these systems. The study of the topics introduced above has been classified under the rubric *Systems biology* Ideker et al. [2001].

In general, the goal of Systems biology is the elucidation of the emergence of complex behaviors in biological systems from relatively simple constituents and the interactions among them. Examples of such biological systems include metabolic networks, cell signaling networks, population dynamics, among many others. The focus of this work will lie on models of GRNs. However, this does not imply that the underlying mathematical methods used here are necessarily restricted to these biological systems.

In such a holistic approach, the entities of the biological system must be abstracted in a suitable way. In the context of GRNs, this could mean for example that we do not distinguish between the many ways genes are regulated by transcription factors, but rather, that we simply consider their capacity for activating or inhibiting the activity of their own or other genes.

[1] This does not hold in general; there are some exceptions neglected in this introduction (see e.g. Chai and Gleeson [2018]).

In this way, molecular interactions are abstracted in a mathematical formalism. With such an abstraction, the aim is both to understand the essential mechanisms characterizing the system, and ultimately to build realistic computer models of the system.

What are the challenges in modeling GRNs? In the concrete cases, it is not always clear how far we can go with these abstractions. This is reflected by the presence of several modeling approaches that are often used in concert.

In principle, it is possible to build *quantitative models*, i.e. models giving precise numerical predictions of GRN behavior, with a set of differential equations. If the number of molecules involved in the modeled network is small, randomness can be taken into account additionally using stochastic models. Such models can represent very detailed descriptions of the modeled regulatory networks. Indeed, such models have been successfully applied to small regulatory networks (see e.g. Barkai and Leibler [1997], Tyson [1991], Tindall et al. [2012], De Jong [2004]). However, from a practical perspective this approach is usually hampered by the many unknown kinetic parameters and mechanistic details of the majority of modeled biological systems. Furthermore, it is not always necessary to have quantitative information about reaction mechanisms and kinetic constants to understand GRN function [Szallasi et al., 2006, p. 125]. Therefore, *qualitative models* are considered. Qualitative models of GRNs are designed to capture the properties of the modeled GRN that are invariant over a range of different reaction mechanisms and values of kinetic constants [Szallasi et al., 2006, p. 125]. Therefore, these models can be applied to GRNs about whose details less is known. Additionally, they can provide information regarding which properties of a system are important or essential, and which are not.

0.2. Contributions of this thesis

This thesis is concerned with qualitative models of gene regulatory networks. More specifically, we will consider and compare the following types of models:

(1) interaction graphs
(2) Boolean networks
(3) models based on differential equations
(4) discrete abstractions of differential equations

An interaction graph of a GRN is a structural description of the GRN describing the possible interactions among the components of the network. In contrast to other kinds of models considered here, interaction graphs contain no explicit notion of time. There are several ways to refine these models to dynamic models, i.e. models incorporating a notion of time. So called Boolean networks assume that genes can have only two states, "active" and "inactive". Thus, Boolean networks can describe one or more possible sequences of activation and inactivation of the components of the GRN over time. Another possibility is to build a parametrized model of differential equations. Since the biochemical reaction mechanisms in the GRN are normally not known in detail, some assumptions or simplifications are usually made. In this thesis we consider differential equations based on Hill kinetics. Such models are widely used Goutelle et al. [2008], Santillán [2008], Griffith [1968a,b], Deuflhard and Röblitz [2015], Aldridge et al. [2006], Lewis [2003], Özbudak and Lewis [2008], Lange et al. [2018], Krumsiek et al. [2010], Wittmann et al. [2009]. Another alternative not treated here is piecewise linear differential equations Edwards et al. [2001], Glass and Kauffman [1973], Gouzé and Sari [2002], Mestl et al. [1995]. The last class of models we consider is that of models based on discrete abstractions of differential equations. They build on the reasoning of models consisting of differential equations. These models partition the phase space into regions and describe restrictions on the transitions between these regions. They are constructed in such a way, that possible transitions between the regions can be computed in the absence of precise numerical information on the parameters and exact kinetics. Therefore, they are treated as independent modeling framework here.

The goal of this thesis is to compare these four different modeling formalisms from a theoretical point of view. This comparison is important for two reasons.

(1) It constitutes an approach to validate the predictions of different types of models in a general way. The assumption is that the properties present across different modeling formalisms are likely to be robust predictions even if our models lack the full detail of the modeled object.
(2) Information from different models is synergistic; results from simple models can be transferred to more complex models, helping to simplify or supplement their analysis. Since the feasibility and computational cost of the analysis in different model classes can vary greatly, this can potentially be very advantageous, even allowing the analysis of certain classes of models which otherwise would not be feasible.

There are already several works comparing different qualitative modeling approaches. The relation between interaction graph and the corresponding Boolean networks and models based on differential equations is an extensively studied field. Much of the research has focused on linking the presence of structural properties – such as positive or negative cycles in the interaction graph – to dynamical properties (e.g. number of fixed points, attractors) Thomas and Kaufman [2001a], Remy et al. [2008], Soulé [2003], Richard and Comet [2010], Thomas [1981], Snoussi [1998], Kaufman et al. [2007], Thomas and Kaufman [2001b], Tonello et al. [2019]. For models based on differential equations another approach exists. Discrete abstractions of differential equations constitute an alternative approach to linking structure and dynamics. These models are constructed directly from interaction graphs and therefore inherently constitute a relation between these modeling frameworks. We will make three contributions in the effort to relate different modeling frameworks in this thesis:

(1) The idea of abstracting the dynamics of individual models to relate structure and dynamics constitutes one of the main approaches in this thesis. We will apply this approach to Boolean models.

This will allow us to compare groups of models in different modeling frameworks (Boolean networks and differential equations) having the same structure. These ideas have been partially published Schwieger and Siebert [2017], but the results here will be presented in a more systematic way and include several new results.

(2) We will show that certain state transition graphs of discrete abstractions of differential equations can be considered asynchronous Boolean networks. This work is published in Schwieger and Siebert [2018].

(3) The other approach taken in this thesis concerns individual Boolean models and parametrized families of ODEs. To construct ODE models systematically from Boolean models, several automatic conversion algorithms have been proposed Wittmann et al. [2009], Krumsiek et al. [2010]. We will consider several closely related algorithms and prove that certain invariant sets are preserved during the conversion. These results are published in Schwieger et al. [2018].

Despite the rather theoretical question this thesis tries to answer there are surprisingly many potential applications of the results here such as

- network reduction of ODE models with Hill kinetics in Chapter 2,
- network inference and analysis of Boolean model pools in Chapter 4.

The final chapter will therefore be dedicated to several ideas for applications with respect to experiment design. These ideas can be used in other contexts as well. They should serve as a primer for future research.

0.3. Structure of this thesis

In Chapter 1 we introduce two modeling frameworks: dynamic models based on systems of ordinary differential equations (ODEs) and Boolean models. After a brief introduction and motivation of the construction of such ODE models, we turn our attention **in Chapter 2** to the comparison of these two classes of models. Namely, we focus on a specific assignment of ODE models to Boolean networks introduced in Krumsiek et al. [2010]. We show that we can shift a part of the analysis of the ODE model to the corresponding Boolean network by considering so-called trap spaces – parts of the state space where the values of certain components do not change. The resulting family of ODE-systems can vary along a large number of parameters and can already be considered a qualitative model. Therefore, it serves as an indicator as to which properties of Boolean models might be expected to be relatively robust predictions. The results of this chapter are mainly published in Schwieger et al. [2018]. For Chapter 2, little knowledge of Boolean networks is necessary.

Therefore, before continuing with the comparison of continuous and Boolean models, **in Chapter 3** we give an overview of the mathematical tools we use in the investigation of Boolean models. The most important constructions introduced in Chapter 3 are two types of quotient graphs: those relying on the update of the Boolean network and those constructed using projection maps. They will be used frequently in the subsequent chapters. Intuitively, they can be understood as a way of abstracting the behavior of a Boolean model, allowing the joint analysis of sets of Boolean models. The tools presented in this chapter are introduced for application in this thesis, but they also represent a succinct overview of methods useful in other contexts.

In Chapter 4 we investigate different aspects of abstractions of models. We start by introducing discrete abstractions of differential equations using methods from *qualitative reasoning* Kuipers [1984], Eisenack [2006]. Historically, these ideas go back to work by Kuipers in artificial intelligence Kuipers [1984]. By partitioning phase space into a finite number of regions, a transition graph describing possible behaviors of the system is constructed. In Chapter 4 we will only consider a very elementary way of partitioning the phase space based on the signs of derivatives. We will compare this method of abstracting continuous models to one of the abstraction methods for Boolean networks introduced in Chapter 3. This leads us to the first main result of Chapter 4: Abstraction methods in Boolean networks and continuous models lead to very similar state transition graphs.

When considering discrete abstractions of differential equations, the resulting state transition graph can be constructed by purely logical rules. In particular, no explicit knowledge about the corresponding differential equations is necessary. Therefore, they can be treated as an independent modeling formalism (see e.g. [Szallasi et al., 2006, p. 125]). However, by construction, their relation to models based on differential equations and the underlying interaction graphs is well understood. In Section 4.2 we mirror this approach to obtain the analogous result for Boolean networks. Afterwards, we show that, under the assumptions made in this thesis, we can unify both approaches and consider them mathematically as a specific class of Boolean networks – which we call skeletons of such transition graphs. This result concludes our comparison of different modeling frameworks. The results of Chapter 4 have been partially published in Schwieger and Siebert [2018, 2017].

The last chapter – Chapter 5 – can serve as an outlook for future work. In this chapter, we discuss several methods of comparison of different Boolean networks. We do this in order to develop applications of the results in Chapter 4. However, as in Chapter 3, due to the generality of these results, they may well be of interest in very different contexts. In the context of this thesis, we use the methods of comparing different Boolean networks combined with the results in Chapter 4 to develop criteria for distinguishing different interaction graphs. These criteria are based on states, transitions or sequences of states. We give different formulations of these problems using quantified Boolean formulas, finite automata and algebraic geometry.

CHAPTER 1

Boolean models and ODE-models of gene regulatory networks

Contents

1.1. General notations **8**

1.1.1. Graph theory 8

1.1.2. Boolean expressions 10

1.1.3. Notation used in the context of Ordinary differential equations and dynamical systems 11

1.2. ODE models of regulatory networks **13**

1.2.1. Mass action kinetics 13

1.2.2. Hill kinetics 14

1.3. Boolean networks **16**

1.3.1. Synchronous Boolean network 17

1.3.2. Asynchronous Boolean network 18

1.3.3. Dynamics of Boolean networks 19

1.3.4. Structure of Boolean networks 20

In this chapter we will introduce very briefly the basic mathematical background and standard notation used throughout this thesis. It should be understood as a motivation and a collection of essential definitions and notations used in this thesis. For the reader unfamiliar with this topics or interested in a more in-depth introduction we will give references to more comprehensive introductions. We will introduce two types of modeling frameworks: Boolean models and ODE models. We start by reviewing some basic definitions from mainly logic and graph theory in Section 1.1. The definitions from logic will be used in the context of *Boolean models* introduced in Section 1.3. Graph theory will appear in the context of the analysis of the structure and dynamics of Boolean models. In this context also notions such as *steady states, trap sets* and *attractors* are introduced, which will be used throughout this thesis in the context of *Boolean networks*. Graph theory will also appear in the context of ordinary differential equations in Section 4.1 as a tool to investigate abstractions of model pools.

Afterwards, in Section 1.2 we will introduce continuous models of GRNs based on ordinary differential equations. To give a taste of how these models are derived and constructed we explain briefly the *mass action law* and the *Hill kinetics*. The latter one will be extensively used in Chapter 2.

In Section 1.3 we introduce Boolean models of GRNs. We will consider two types of Boolean networks: *synchronous Boolean networks* and *asynchronous Boolean networks*. For both types of Boolean networks we will explain what we mean by its dynamics and structure. For this purpose the interaction graph and the state transition graph of a Boolean network is introduced.

1.1. General notations

We start by introducing the following conventions for sets:

Definition 1.1. For a set A we denote with $|A|$ its cardinality and with $\mathscr{P}(A)$ the powerset of A, i.e. the set of all subsets of A.

We use the symbol $\mathbb{N}$ to denote the natural numbers not including zero. For the set given by the natural numbers until $n \in \mathbb{N}$, i.e. $\{1, \dots, n\}$ we write $[n]$.

Next, we introduce some notation mainly used for Boolean vectors, but also sign vectors in Section 4.1. Since we will consider later graphs whose nodes are Boolean vectors, this will come handy to the describe the edges of such graphs.

Definition 1.2. For two elements in a Cartesian product of n sets, i.e. $v, w \in \prod_{i=1}^{n} A_i$ with A_i any set, we define

$$\begin{aligned} \operatorname{diff}(v,w) &:= \big\{i \in [n] | v_i \neq w_i\big\}, \\ \operatorname{comm}(v,w) &:= [n] \backslash \operatorname{diff}(v,w). \end{aligned}$$

For a tuple $e = (v,w)$ we also write $\operatorname{diff}(e)$ instead of $\operatorname{diff}(v,w)$. And for a set of tuples E we write $\operatorname{diff}(E)$ for the union $\bigcup_{e \in E} \operatorname{diff}(e)$.

In other words the set $\operatorname{diff}(v,w)$ describes the set of indices where v and w differ from each other. The set $\operatorname{comm}(v,w)$ is the set of indices, where v and w do not differ. Typically in the above definition we will choose $A_i = \{0,1\}$, $i \in [n]$ such that $\prod_{i=1}^{n} A_i = \{0,1\}^n$. Sometimes, we will replace however the set $\{0,1\}$ with $\{-1,1\}$. Therefore, we gave a more general definition.

1.1.1. Graph theory.

We are going to consider models of regulatory networks which can be described as directed graphs or at least some properties of these models can be described as directed graphs. Therefore, we need some basic definitions of graph theory.

A *directed graph* G is an ordered pair of disjoint sets (V,E) such that the elements in E are identified with elements in $V \times V$. The set V is called the set of *nodes* and E is called the set of *edges*. If G is a graph then $V(G)$ is the set of nodes of G and $E(G)$ is the set of edges. Let $G = (V,E)$ be any directed graph. For $(v,w) \in E$ we also write $v \to w$ if it is from the context clear which graph is meant. Edges of the form (v,v) in G are called *loops*. For two graphs $G' = (V',E')$ and $G = (V,E)$ we say G' is a subgraph of G if $V' \subseteq V$ and $E' \subseteq E$. Especially, therefore every graph is a subgraph of itself. We introduce the following symbols:

Definition 1.3. The set of *predecessors* $\operatorname{pred}(v)$ and the set of *successors* $\operatorname{succ}(v)$ of a node $v \in V$ are defined as

$$\begin{aligned} \operatorname{pred}(v) &:= \big\{w \in V | (w,v) \in E\big\}, \\ \operatorname{succ}(v) &:= \big\{w \in V | (v,w) \in E\big\}. \end{aligned}$$

With $\deg^{+}(v)$ we denote its *outdegree*, the number of its successors, and with $\deg^{-}(v)$ its *indegree*, the number of its predecessors.

Next, we introduce notations related to the reachability of nodes.

Definition 1.4 ([Bollobás, 2013, p.4])**.** A *(directed) path* is a directed graph $P = (V(P), E(P))$ of the form $V(P) = \{x_1, \ldots x_{l+1}\}$ with $x_i \rightarrow x_{i+1}$, $i \in [l]$ and no other edges. The length of the path is l – i.e. the number of edges.

In the above definitions paths of length zero are not excluded. Therefore, a single node is by definition always a directed path.

Definition 1.5. We write $v \rightsquigarrow w$ for $v, w \in V$ iff there is a directed path from v to w in G. We also say in this case w is *reachable* from v. We call the graph $(V, \rightsquigarrow)$ the *transitive closure* of G.

Again, the above definition implies that every node is reachable from itself.

Definition 1.6 ([Bollobás, 2013, p.5])**.** A *walk* in a directed graph $G = (V, E)$ is a finite sequence of nodes in G – let say $(x_i)_{i \in [l+1]}$ such that $x_i \rightarrow x_{i+1}$ for $i \in [l]$. We call l the length of the walk. A walk whose edges are distinct is called *trail.* A *circuit* is a trail $(x_i)_{i \in [l]}$ such that its first and last node coincide, i.e. $x_1 = x_l$. A walk $W = (x_i)_{i \in [l+1]}$ with $l \geq 3$, $x_1 = x_{l+1}$, and the nodes x_i, $i \in [l]$ are distinct from each other is called *cycle.*

We also introduce some not so common notions which will be used mainly in the context of Boolean networks (introduced in Section 1.3) and their state transition graphs.

Often we are interested in regions of these graphs from which no walk are leading out. Such regions are called *trap sets.*

Definition 1.7 ([Klarner, 2015, p. 78])**.** For a directed graph $G = (V, E)$ we call $T \subseteq V$ a *trap set* if there are no edges from T to $V \backslash T$ in G – i.e. $\big[T \times V \backslash T\big] \bigcap E = \emptyset$.

The complements of trap sets are called no-return sets. These sets are characterized by the fact that no path that left it can enter it again.

Definition 1.8. For a directed graph $G = (V, E)$ we call $T \subseteq V$ a *no-return set* if there are no edges from $V \backslash T$ to T in V– i.e. $\big[V \backslash T \times T\big] \bigcap E = \emptyset$ or differently expressed a set whose complement is a trap set.

Often several trap sets are nested. In this case, of special interest are inclusion-wise minimal trap sets. These sets are called *attractors*. In many models of regulatory networks these regions are of special interest since they capture a possible long-term behavior of the modeled system.

Definition 1.9 (Klarner and Siebert [2015])**.** For a directed graph $G = (V, E)$ we call a subset of V *attractor* if it is an inclusion-wise minimal trap set. An attractor consisting only of one node is called *steady node* (or in the context of Boolean networks *steady state*)

When talking about attractors (or more general trap sets) its often useful to know from which nodes the attractor is reachable. For this purpose the notions weak and strong basin of attraction are introduced. The weak basin of attraction is the set of nodes from which at least one path leads to the attractor. For nodes in the strong basin all sufficiently long paths starting in these nodes lead eventually into the corresponding attractor. We formalize this in the following definition.

Definition 1.10 (Klarner et al. [2018]). For a trap set of a directed graph $G = (V, E)$ we call its *weak basin of attraction* the set of nodes from which exists a directed path to the trap set. The *strong basin of attraction* consists of all nodes such that from all nodes reachable from it there is a path to the trap set. More formally: For a trap set $T \subseteq V$ we define:

$$\begin{aligned} \mathrm{WeakBasin}(T) &= \{x \in V | \exists t \in T : x \rightsquigarrow t\}, \\ \mathrm{StrongBasin}(T) &= \{x \in V | \forall y \in V : (x \rightsquigarrow y) \Rightarrow (\exists t \in T : y \rightsquigarrow t)\}. \end{aligned}$$

We also remind the reader to the definition of a graph homomorphism:

Definition 1.11 (see e.g. [Hahn and Tardif, 1997, Definition 2.1]). A *graph homomorphism* f from a directed graph $G = (V(G), E(G))$ to a graph $H = (V(H), E(H))$ is a function from $V(G)$ to $V(H)$ such that $(v, w) \in E(G) \Rightarrow (f(v), f(w)) \in E(G)$ holds. If a graph homomorphism f from G to H is a bijection whose inverse function f^{-1} is a graph homomorphism as well, then f is a *graph isomorphism*.

1.1.2. Boolean expressions. To describe the edges in the transitions graphs introduced in the context of Boolean networks and qualitative differential equations in Section 4.1 we use Boolean expressions. A Boolean expression is composed of Boolean operators.

Definition 1.12. We use the following Boolean operations: $\wedge$ for the Boolean and, $\vee$ for the Boolean or, $\neg$ for the negation, $\rightarrow$ for an implication, $\leftrightarrow$ for Boolean equivalence and $\oplus$ for xor.

A Boolean expressions is then defined as follows:

Definition 1.13 ([Crama and Hammer, 2011, p. 10]). Given a finite collection of Boolean variables $\{x_1, \dots, x_n\}$, a *Boolean expression* (or *Boolean formula*) in the variables $x_1, \dots, x_n$ is defined recursively as:
(1) The constants $0, 1$ and the variables $x_1, \dots, x_n$ are Boolean expressions in $x_1, \dots, x_n$.
(2) If α and β are Boolean expressions in $x_1, \dots, x_n$, then $\alpha \vee \beta$, $\alpha \wedge \beta$ and $\neg\alpha$ are Boolean expressions in $x_1, \dots, x_n$.
(3) Every Boolean expression is formed by finitely many applications of the rules (1) and (2).

We use capital $\bigwedge$ if we consider the conjunction of a set of Boolean expressions. For example $\bigwedge_{i \in [n]} a_i = a_1 \wedge \dots \wedge a_n$, $a \in \{0, 1\}^n$. We use the convention that the conjunction of the empty set is always true. For example $\bigwedge_{i \in \emptyset} a_i = 1$ for all $a \in \{0, 1\}^n$. Similar we proceed for $\bigvee$. But here we use the convention that the disjunction over an empty set of indices is always false. I.e. for example $\bigvee_{i \in \emptyset} a_i = 0$.

Since the above Boolean operations can be expressed in the field $\mathbb{F}_2$ (the Galois field of two elements) the set $\{0, 1\}^n$ forms a vector space. We extend all Boolean operations introduced above to vectors over $\{0, 1\}^n$ in a component-wise way. For vectors $\begin{pmatrix} s_1 & \dots & s_n \end{pmatrix} \in \{0, 1\}^n$ we also just write

$s_1 \dots s_n$. For example instead of $\begin{pmatrix}0 & 1 & 0\end{pmatrix}$ we just write 010. We can define also a metric on the set of Boolean vectors $\{0,1\}^n$:

Definition 1.14. The *Hamming distance* of two vectors $v, w \in \{0,1\}^n$ is defined as the cardinality of the set $\operatorname{diff}(v, w)$.

We also introduce here the following notation, which will be used mainly in Chapter 4.

Definition 1.15. For $a, b \in \{0,1\}$ we define $a \oplus^1 b := a \oplus b$, $a \oplus^0 b = 0$ and $a \oplus^{-1} b := a \leftrightarrow b$.

We use notations like $\alpha(x_1, \dots, x_n)$ to denote Boolean expressions in the variables $x_1, \dots, x_n$. Since every Boolean expression induces a Boolean function $\{0,1\}^n \to \{0,1\}$ we identify the Boolean expression α with the functions it induces and do not distinguish between these objects when not otherwise stated. For example we write $f(x_1, x_2) = x_1 \wedge x_2$ for the function $f : \{0,1\}^2 \to \{0,1\}, (x_1, x_2) \mapsto \begin{cases} 1 & \text{if } x_1 \wedge x_2, \\ 0 & \text{otherwise.} \end{cases}$ We denote with $\mathbb{B}(n, m)$ the set of Boolean functions from $\{0,1\}^n$ to $\{0,1\}^m$, $n, m \in \mathbb{N}$.

Definition 1.16. A *literal* is a Boolean expression of the form x or $\neg x$ for a Boolean variable x. A *term* T is a Boolean expression in $x_1, \dots, x_n$ of the form

$$T = \bigwedge_{i \in A} x_i \wedge \bigwedge_{j \in B} \neg x_j, A, B \subseteq [n], A \bigcap B = \emptyset.$$

A *clause* C is a Boolean expression in $x_1, \dots, x_n$ of the form

$$C = \bigvee_{i \in A} x_i \vee \bigvee_{j \in B} \neg x_j, A, B \subseteq [n], A \bigcap B = \emptyset.$$

Let us also recall the definitions of disjunctive and conjunctive normal forms:

Definition 1.17 ([Crama and Hammer, 2011, Def. 1.10]). A *disjunctive normal form* (DNF) is an expression of the form $\bigvee_{k=1}^m T_k$, where T_k, $k \in [m]$ are terms. A *conjunctive normal form* (CNF) is an expression of the form $\bigwedge_{k=1}^m C_k$ where C_k, $k \in [m]$ are clauses.

1.1.3. Notation used in the context of Ordinary differential equations and dynamical systems. With $C(X, Y)$ we denote the space of continuous functions from X to Y. More general, with $C^k(X, Y)$ we denote the space of functions $f : X \to Y$ which have continuous derivatives $f^{(1)}, \dots, f^{(k)}$. We use a dot to denote for a function $x : \mathbb{R} \to \mathbb{R}^n$ its derivative, i.e. $\dot{x}$. For a function $f : \mathbb{R}^n \to \mathbb{R}, x \mapsto f(x)$ we denote with $\partial_j f$ the partial derivative of f with respect to x_j. With $\nabla f = \begin{pmatrix}\partial_1 f & \dots & \partial_n f\end{pmatrix}$ we denote the gradient of f. Finally, for a function $f : \mathbb{R}^n \to \mathbb{R}^m$ we define J_f to be the Jacobian matrix of f.

We will use also a few notions from dynamical systems in this thesis.

Definition 1.18 ([Teschl, 2012, p. 187]). A *dynamical system* is a tuple (G, M, T), where $(G, +)$ is a semigroup with identity element e acting on a set M and T is a map

$$\begin{aligned} T : U \subset (G \times M) &\to M, \\ (g, x) &\mapsto T_g(x), \end{aligned}$$

such that

$$T_g \circ T_h = T_{g+h}, T_0 = \mathbb{I},$$

where $\mathbb{I}$ is the identity. The map T is called the *evolution function* of the dynamical system. The set M is called *state space* (or *phase space*). If we consider T for a fixed $x \in M$, we call the induced map, the *flow* through x.

Definition 1.19. We define the *orbit (or trajectory) through* $x \in M$ as the set

$$\gamma(x) := \{T_g(x) | g \in I(x)\} \subseteq M$$

with $I(x) := \{g \in G | (g, x) \in U\}$. A point whose orbit consists of a single point, we call *fixed point.*

Definition 1.20. For a dynamical system (G, M, T) a set $A \subseteq M$ is called an *invariant* set if

$$\forall x \in A \forall g \in G : T_g(x) \in A$$

is satisfied.

In this thesis we consider real dynamical systems, where the semigroup in Definition 1.18 is given by the positive real numbers, i.e. $(G, +) = (\mathbb{R}_{\geq 0}, +)$, and discrete dynamical systems, where $(G, +) = (\mathbb{Z}_{\geq 0}, +)$. Real dynamical systems can arise from an ODE-system of the form

$$\begin{aligned} \dot{x} &= f(x), \\ x(0) &= x_0, \end{aligned}$$

where $f \in C^k(M, \mathbb{R}^n)$, $k \geq 1$, and $M \subseteq \mathbb{R}^n$ is open. Indeed, there is a unique maximal integral curve ϕ_x at every point $x \in \mathbb{R}^n$, defined on some neighborhood I_x of x. This allows to define the flow of the above differential equation to be the map

$$\begin{aligned} T : U &\to M, \\ (t, x) &\mapsto \phi_x(t), \end{aligned}$$

where $U = \bigcup_{x \in M} I_x \times \{x\} \subseteq \mathbb{R} \times M$ and x is fixed. A fixed point x of T is called *stable* if for any neighborhood $V(x) \subseteq I_x$ there exists a second neighborhood $V'(x) \subseteq V(x)$ such that any solution starting in $V'(x)$ remains in $V(x)$ for $t \geq 0$. A fixed point x is called *asymptotically stable* if it is stable and if there is a neighborhood $V(x)$ such that

$$\forall y \in V(x) : \lim_{t \to \infty} |T_t(y) - x| = 0$$

holds. For details we refer to [Teschl, 2012, p. 192].

1.2. ODE models of regulatory networks

Continuous dynamic modeling is widely used to model regulatory networks Aldridge et al. [2006], Karlebach and Shamir [2008]. Here we will solely focus on continuous models built with ODEs. The unknown variables in these ODEs represent production and consumption rates of individual biomolecular species. We give a very brief introduction to the construction of these models and introduce as well the Hill kinetics, which will be used later. For a more detailed treaty we refer to Klipp et al. [2008], Deuflhard and Röblitz [2015].

ODE models of regulatory networks are based on the law of mass action kinetics.

1.2.1. Mass action kinetics. The law of *mass action kinetics* states that rates of reactions are proportional to the concentrations of the reactants to the power of the molecularity [Klipp et al., 2008, p. 141]. For example, if we consider a reversible reaction in which the molecules A and B react to C, e.g.

$$\mathrm{A} + \mathrm{B} \underset{k_-}{\overset{k_+}{\rightleftharpoons}} 2\,\mathrm{C},$$

then the so-called reaction rate[1] v is given by $v(A,B,C) := k_+ A \cdot B - k_- C^2, k_+, k_- \in \mathbb{R}_{>0}$, where we identified the names of the molecules A, B and C with its concentrations. The constants k_+, k_- are the respective proportionality factors. The corresponding ODE-system has the form:

$$\begin{aligned}\dot{A} &= -v(A,B,C),\\ \dot{B} &= -v(A,B,C),\\ \dot{C} &= 2 \cdot v(A,B,C).\end{aligned}$$

In general, for a reaction of the form

$$\sum_{i=1}^{n} \alpha_i \mathrm{A}_i \underset{k_-}{\overset{k_+}{\rightleftharpoons}} \sum_{i=1}^{m} \beta_i \mathrm{B}_i, \alpha_i, \beta_i \in \mathbb{N}_{>0}$$

we obtain the reaction rate

$$v = k_+ \prod_{i=1}^{n} A_i^{\alpha_i} - k_- \prod_{i=1}^{m} B_i^{\beta_i}.$$

In most models of regulatory networks enzymatic reactions need to be taken into account. The simplest case is a one-substrate reaction without backward reaction without effectors:

$$\mathrm{E} + \mathrm{S} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \mathrm{ES} \xrightarrow{k_2} \mathrm{E} + \mathrm{P}, \tag{1.2.1}$$

where the letter E stands for a free enzyme, S for a substrate, ES for an enzyme-substrate complex. The reversible reaction on the left describes the formation of the enzyme-substrate complex ES. The irreversible part on the right describes the release of the product P from the enzyme E. To simplify such systems it is often assumed that the first reversible reaction in (1.2.1) is in a so-called quasi-steady state. The argument is that the reaction rates

[1] Change of concentration per time

k_1 and k_{-1} are much bigger than k_2 and therefore there is always enough time to maintain the equilibrium for the first reaction (see Segel [1988] for details).

Enzymes enable reactions to occur on lower temperatures in the cell. Since their production rate can be up- or down-regulated and since their functionality can be altered by the binding of ligands, they are the central building blocks of regulatory networks.

1.2.2. Hill kinetics. When knowledge about the chemical reactions of enzymes is only qualitative, often so-called Hill functions are assumed. They are frequently found within models of gene expression since the late sixties Griffith [1968a,b]. They were introduced in 1910 by A. Hill Hill [1910]. The motivation for their usage was the binding of *ligands*[2] to a protein. Often a protein has several identical binding sites. For example hemoglobin has four binding sites for oxygen [Klipp et al., 2008, p. 154]. In many instances interactions between these binding sites occur. To illustrate this we consider the case of two binding sites (see Santillán [2008] for the general case):

Assume two ligands S bind *cooperatively* a receptor E. The word *cooperatively* means that the binding of the first ligand to an available site increases the chances that another ligand binds to a second empty nearby site. For our example we assume that the receptors E have two identical binding sites for the ligand S. The reaction for the binding of the first ligand has the form

$$\mathrm{E} + \mathrm{S} \underset{k_-}{\overset{2k_+}{\rightleftharpoons}} \mathrm{E_S}.$$

We inserted a factor 2 into the forward reaction rate to take into account that there are two binding sites for the molecule S. The concentrations at the equilibrium points are related by

$$E \cdot S = \frac{K}{2} \cdot E_S,$$

with $K = \frac{k_-}{k_+}$. The constant K is called the *dissociation constant*. Because of the assumed cooperative binding sites for the reaction of the second ligand either k_+ or k_- decreases after the binding of the first ligand. In both cases the corresponding dissociation constant diminishes:

$$\mathrm{E_S} + \mathrm{S} \overset{\frac{2 \cdot K}{k_c}}{\rightleftharpoons} \mathrm{E_{2S}},$$

with $k_c > 1$ describing the degree of cooperativity. The concentrations at equilibrium are related now by

$$E_S \cdot S = \frac{2K}{k_c} \cdot E_{2S}.$$

Assuming a constant number of receptors, i.e. $E + E_S + E_{2S} = E_{\text{tot}}$, it is possible to obtain from the above equations the following formulas for the

[2] Any molecules that bind to protein is called ligand [Klipp et al., 2008, p. 154].

fractions of receptors which have ligands bound to their site[3].

$$\begin{aligned} \frac{E}{E_{\text{tot}}} &= \frac{1}{1+2\cdot\frac{S}{K}+k_c\cdot(\frac{S}{K})^2}, \\ \frac{E_{2S}}{E_{\text{tot}}} &= \frac{k_c\cdot\left(\frac{S}{K}\right)^2}{1+2\frac{S}{K}+k_c(\frac{S}{K})^2}. \end{aligned} \tag{1.2.2}$$

With the substitution $x := \sqrt{k_c}\cdot\frac{S}{K}$ we obtain

$$\begin{aligned} \frac{E}{E_{\text{tot}}} &= \frac{1}{1+2\cdot\frac{1}{\sqrt{k_c}}\cdot x+x^2}, \\ \frac{E_{2S}}{E_{\text{tot}}} &= \frac{x^2}{1+2\cdot\frac{1}{\sqrt{k_c}}\cdot x+x^2}. \end{aligned}$$

For the limit case of very high cooperativity and assuming that the value of x is independent of k_c – i.e. non-diverging x (see [Santillán, 2008, Section 4]) – we obtain:

$$\begin{aligned} \lim_{k_c\to\infty}\frac{E}{E_{\text{tot}}} &= \frac{1}{1+x^2}, \\ \lim_{k_c\to\infty}\frac{E_{2S}}{E_{\text{tot}}} &= \frac{x^2}{1+x^2}. \end{aligned}$$

This arguments can be generalized to a receptor E with n binding sites. In the limit case of high cooperativity we obtain:

$$\begin{aligned} \lim_{k_c\to\infty}\frac{E}{E_{\text{tot}}} &= \frac{1}{1+x^n}, \\ \lim_{k_c\to\infty}\frac{E_{nS}}{E_{\text{tot}}} &= \frac{x^n}{1+x^n}. \end{aligned}$$

These equations are called *Hill equation* or *Hill functions*. In practice often it is unknown how many binding sites a protein possesses and if the binding of ligands is cooperative. Practice has shown however, that the Hill equation

[3] For example the derivation of (1.2.2) is as follows using the above identities:

$$\begin{aligned} & E_S\cdot S = \frac{2K}{k_c}\cdot E_{2S} \\ \Leftrightarrow & (E_{\text{tot}}-E_{2S}-E)\cdot S = \frac{2K}{k_c}\cdot E_{2S} \\ \Leftrightarrow & E_{\text{tot}}\cdot S - E\cdot S = (S+\frac{2K}{k_c})\cdot E_{2S} \\ \Leftrightarrow & E_{\text{tot}}\cdot S - \frac{K}{2}\cdot E_S = (S+\frac{2K}{k_c})\cdot E_{2S} \\ \Leftrightarrow & E_{\text{tot}}\cdot S - \frac{1}{S}\cdot\frac{K}{2}\cdot S\cdot E_S = (S+\frac{2K}{k_c})\cdot E_{2S} \\ \Leftrightarrow & E_{\text{tot}}\cdot S - \frac{1}{S}\cdot\frac{K}{2}\cdot\frac{2K}{k_c}\cdot E_{2S} = (S+\frac{2K}{k_c})\cdot E_{2S} \\ \Leftrightarrow & E_{\text{tot}}\cdot S = (S+\frac{2K}{k_c}+\frac{1}{S}\cdot\frac{K^2}{k_c})\cdot E_{2S} \\ \Leftrightarrow & \frac{E_{2S}}{E_{\text{tot}}} = \frac{S}{(S+\frac{2K}{k_c}+\frac{1}{S}\cdot\frac{K^2}{k_c})} = \frac{S^2}{(S^2+S\cdot\frac{2K}{k_c}+\frac{K^2}{k_c})} = \frac{\frac{k_c}{K^2}S^2}{(\frac{k_c}{K^2}S^2+2\frac{S}{K}+1)} \end{aligned}$$

describes the empirical data with good precision in many instances [Alon, 2007, p. 13].

In summary, ODE models of regulatory networks are based on mass action kinetics. Consequently, for the construction of ODE models many proportionality factors need to be estimated. Furthermore, many reactions are catalyzed by enzymes. Their functionality can be changed by ligands in many instances. When knowledge is not detailed enough, Hill kinetics can be used to model such effects. Nevertheless, this requires for example knowledge about the number of bindings sites for the ligands involved. Consequently, the use of ODE models requires sufficient knowledge about mechanistic details and kinetic parameters of the underlying systems.

1.3. Boolean networks

Boolean networks (BNs) constitute a family of models of regulatory networks, which can be used if only little data is available or systems become relatively large and a simulation with ODE systems would be too costly. Nevertheless, they lead in many instances to useful predictions Wang et al. [2012].

Historically, the use of Boolean models in systems biology goes back to Stuart Kauffman (1939-), René Thomas (1928 - 2017) and Motoyosi Sugita (1905-1990) who started to use Boolean algebra in the 1960s and the 1970s to describe gene regulatory networks Abou-Jaoudé et al. [2016].

Stimulated by François Jacob and Jacques Monod who studied bacterial gene regulation Jacob and Monod [1961], several researchers proposed to model regulatory circuits with Boolean algebra. Mitoyosi Sugita was the first among them proposing in 1963 an explicit modeling of bacterial genetic circuits with symbolic logic Abou-Jaoudé et al. [2016], Sugita [1963].

A few years later in 1969 Stuart Kauffman investigated the origin of life and used random Boolean networks to investigate how generic gene regulatory networks organize themselves Kauffman [1969]. He showed that sparse random Boolean networks exhibit very similar properties than regulatory networks of existing organisms. He proposed that cell types correspond to dynamical attractors in BNs and that cell differentiation can be related to the transition to these attractors.

In the same time René Thomas started to use Boolean algebra to model more specific biological regulatory circuits Thomas [1973], Thomas et al. [1976], Thomas [1978].

From a practical point of view BNs have the advantage, that they can be relatively effectively analyzed with methods from model checking and other formal methods Baier and Katoen [2008], Klarner [2015], Monteiro et al. [2014], Batt et al. [2005]. Several Software-tools have been developed for this purpose Klarner et al. [2016], Gonzalez et al. [2006], Videla et al. [2017], Streck [2015]. These methods allow to investigate some properties of models with several hundreds of components Saez-Rodriguez et al. [2007], Samaga et al. [2009], Schlatter et al. [2011, 2009], Abou-Jaoudé et al. [2016].

Formally a BN consists of a set of Boolean variables and a set of logical formulas describing the behavior of the system. In this type of model components of the regulatory network are represented by Boolean variables, i.e.

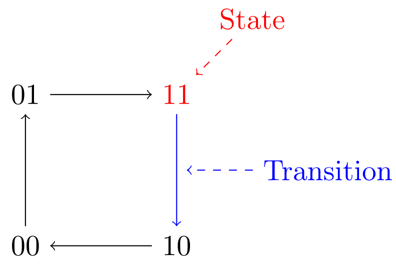


FIGURE 1.3.1. Boolean network with two components $\{1, 2\}$. The nodes of the directed graph representing the behavior of the Boolean network are called states and the edges between the states are called transitions.

they can take either the value zero or one Hopfensitz et al. [2013], De Jong [2004]. The state of a regulatory network with $n \in \mathbb{N}$ components is modeled in this formalism as a Boolean vector in $\{0, 1\}^n$ storing the values of the Boolean variables of the BN. The (possible) behavior of a regulatory network corresponds to trajectories, i.e. sequences of transitions between these states. It can be represented by a directed graph over the set of nodes $\{0, 1\}^n$. Additionally, we require that these transitions can be described by logical expressions. For linking the directed graph representing the dynamics of the Boolean network with the logical rules how the components interact, several mathematical formalisms are in use. We define a general Boolean network here as follows:

Definition 1.21. We call a map from the set of Boolean functions $\mathbb{B}(n, n)$ to the set of directed graphs over $\{0, 1\}^n$ *update-rule*. For $f \in \mathbb{B}(n, n)$ we denote the associated graph under this mapping with $G_{\text{u-rule}}(f) = \big(V_{\text{u-rule}}(f), E_{\text{u-rule}}(f)\big)$, $V_{\text{u-rule}}(f) = \{0, 1\}^n$. For any $f \in \mathbb{B}(n, n)$ we call the tuple $(f, G_{\text{u-rule}}(f))$ a *u-type Boolean network* (u-BN). We call the graph $G_{\text{u-rule}}(f)$ *state transition graph* (STG). The function f we call *update function.* We refer to the set $[n]$ as the *components* of the Boolean network. The nodes of this graph are called *states* and its edges *transitions*.

The update-rule $G_{\text{u-rule}}(\cdot)$ in Definition 1.21 assigns to every Boolean function $f : \{0, 1\}^n \to \{0, 1\}^n$ a directed graph. Sometimes we talk about the dynamics or behavior of a Boolean network. In this case we refer to the structure of its state transition graph.

In the above definition we did not specify the update-rule further. This we will do now. In applications there are mainly two update-rules considered: the synchronous and the asynchronous update.[4] Here we focus mainly on the latter one.

1.3.1. Synchronous Boolean network. A synchronous Boolean network is a Boolean network whose STG consists only of states with outdegree one. The successor of a state is given by the image of the Boolean function inducing the BN. The formal definition is as follows:

[4] These update-rules can also be generalized to probabilistic Boolean networks (PBNs) Liang and Han [2012], Shmulevich et al. [2002]. However, these types of Boolean networks are not considered in this work.

Definition 1.22. A *synchronous Boolean network* is a tuple $\big(f, G_{\text{sync}}(f)\big)$, where $f \in \mathbb{B}(n,n)$ is a Boolean function and $G_{\text{sync}}(\cdot)$ is an update-rule assigning to any $f \in \mathbb{B}(n,n)$ the directed graph $G_{\text{sync}}(f) = \big(V_{\text{sync}}(f), E_{\text{sync}}(f)\big)$ defined by

$$V_{\text{sync}}(f) := \{0,1\}^n$$

and

$$E_{\text{sync}}(f) := \big\{(s,t) \in V_{\text{sync}}(f) \times V_{\text{sync}}(f) \big| t = f(s) \text{ and } s \neq t\big\}.$$

We call the graph $G_{\text{sync}}(f)$ the *synchronous state transition graph* of f.

We remark that we demand in the above definition that a transition can only exist between two distinct states. Also it is easy to see that the mapping $G_{\text{sync}}(\cdot)$ is injective.

1.3.2. Asynchronous Boolean network. In the state transition graph of a synchronous Boolean network states are updated simultaneously. In an asynchronous Boolean network this is not the case. The state transition graph of an asynchronous Boolean network (ASTG) will capture possible behaviors of a modeled regulatory network. The asynchronous update is believed to be more suitable for models for GRNs (see e.g. Harvey and Bossomaier [1997], Luo and Wang [2013]). In an ASTG a state can have possibly many different successor states. Walks in this graph correspond to possible behaviors of the system. The ASTG is induced by a logical function $f : \{0,1\}^n \to \{0,1\}^n$. To do so we compare each state $s \in \{0,1\}^n$ with its image under f – i.e. $f(s)$. Then for each component for which the image $f(s)$ differs from the state s, we add a transition. For example imagine a three dimensional system where the state 000 is mapped by f to 110. Then the corresponding ASTG would contain the transitions $000 \to 100$ and $000 \to 010$ but not the transition $000 \to 110$. The idea behind this formalism is, that it is very unlikely, that in a continuous time systems all updates would take place in the same time. We give now a formal definition:

Definition 1.23. An *asynchronous Boolean network* is a is a tuple $\big(f, G_{\text{async}}(f)\big)$, where $f \in \mathbb{B}(n,n)$ is a Boolean function and $G_{\text{async}}(\cdot)$ is an update-rule $G_{\text{async}}(\cdot) = \big(V_{\text{async}}(\cdot), E_{\text{async}}(\cdot)\big)$ defined by

$$V_{\text{async}}(f) := \{0,1\}^n$$

and

$$\begin{aligned} E_{\text{async}}(f) := \big\{(s,t) \in V_{\text{async}}(f) \times V_{\text{async}}(f) \big| \big(\{i\} = \operatorname{diff}(s,t) \\ \text{and } f_i(s) = t_i\big)\big\}, \end{aligned}$$

We call $G_{\text{async}}(f)$ the *asynchronous state transition graph* (ASTG) of f.

Note that we allow in the above definition no loops, but only transitions, where one component changes. Again it is easy to see that the mapping $G_{\text{async}}(\cdot)$ is injective.

To simplify our notation we introduce similar to Remy et al. [2008] some notation to express that a Boolean vector is partially negated on a set $A \subseteq [n]$.

Definition 1.24. Let $A \subseteq [n]$ and $v \in \{0,1\}^n$. Define $v^A \in \{0,1\}^n$ component wise like this:

$$v_i^A = \begin{cases} v_i & \text{if } i \notin A \\ \neg v_i & \text{if } i \in A \end{cases}.$$

We extend this definition to transitions.

Definition 1.25. For any $e = (v, v^{\{i\}}) \in \{0,1\}^n \times \{0,1\}^n$, $i \in [n]$ and $A \subseteq [n] \setminus \{i\}$ we write e^A for $(v^A, v^{A \cup \{i\}})$.

We can summarize the above as follows:

Lemma 1.1. A graph $G = (\{0,1\}^n, E)$ is an ASTG (i.e. can be represented by a Boolean function $f : \{0,1\}^n \to \{0,1\}^n$ in the sense of Definition 1.23) if and only if every transition $(v, w) \in E$ consists of states with Hamming distance one (see Definition 1.14) from each other.

Proof. For each state $v \in \{0,1\}^n$ in G we can define a function $f(v)$ component-wise by $f_i(v) = \begin{cases} \neg v_i & (v, v^{\{i\}}) \in E \\ v_i & (v, v^{\{i\}}) \notin E \end{cases}$. □

1.3.3. Dynamics of Boolean networks. Throughout this thesis we use the letter f to denote a Boolean function $\{0,1\}^n \to \{0,1\}^n$, $n \in \mathbb{N}$.

To describe the dynamics of a Boolean network we will make often use of the notions introduced in Section 1.1.1. For example a steady state (see Definition 1.9) of a Boolean network is characterized by the states $x \in \{0,1\}^n$ satisfying $f(x) = x$. The notion of a steady state can be generalized to that of a *trap space*. A trap space is a trap set where some of the components remain fixed, while other are still allowed to change their value. For the formal definition we also introduce projections on the set $\{0,1\}^n$, which will be used in Chapter 2 and investigated further in Section 3.2.1.

We introduce the following notation for projections:

Definition 1.26. For any $\emptyset \neq I \subseteq [n]$ we define the map $\text{proj}^I : \{0,1\}^n \to \{0,1\}^{|I|}, x \mapsto (x_i)_{i \in I}$ and call it *projection*.

Then we define trap spaces as certain preimages of such projections:

Definition 1.27. A subset S of the state space $\{0,1\}^n$ is called a *subspace* of $\{0,1\}^n$ (or more precisely an $I-$*subspace)* if there exists a non-empty set $I \subseteq [n]$ and a state $p \in \{0,1\}^{|I|}$ such that it holds $S = (\text{proj}^I)^{-1}(p)$. Sometimes, we also write $(p, *)$ for the subspace $(\text{proj}^I)^{-1}(p)$, when it is clear from the context which components are meant.
An $I-$subspace which is a trap set (see Definition 1.7) we call $I-$*trap space* (or simply *trap space*).

Trap spaces are useful in the analysis of BNs since they can be computed relatively efficiently in contrast to the more general trap sets Klarner et al. [2015]. To get familiar with the notions introduced in Section 1.1.1 we consider a small example for an ASTG.

Example 1.1. Consider the function $f : (x_1, x_2, x_3, x_4) \mapsto (x_1 \vee \neg x_2, \neg x_1 \vee (x_3 \wedge x_4), \neg x_2, x_1)$. Its ASTG is depicted in Figure 1.3.2. We see that

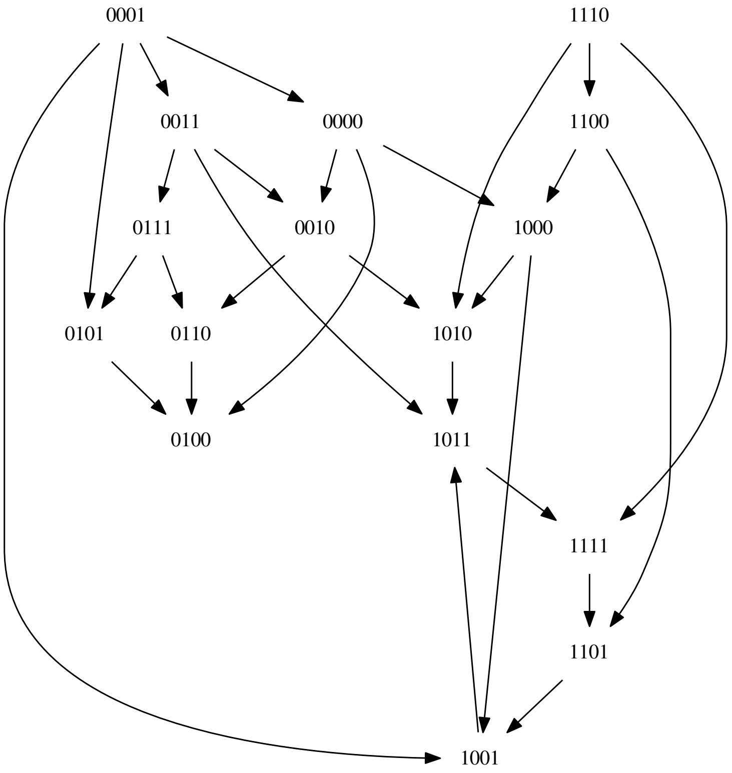


FIGURE 1.3.2. $G_{\text{async}}(f)$ from Example 1.1.

0100 is a steady state (and therefore also an attractor) and for example $T = \{1001, 1011, 1111, 1101\}$ is a trap space where the first and last component is fixed.

1.3.4. Structure of Boolean networks. Next, we describe how we can obtain a structural description of our Boolean network. Such a structural description captures the effect of the components of the regulatory network on each other and can be represented as a graph with the nodes representing components of the network and directed edges representing influences/interactions. Formally it will be another directed graph assigned to a Boolean function $f \in \mathbb{B}(n, n)$. To define what we mean with "influence" we use the notation in Definition 1.24 to define the *discrete derivative* of a Boolean function $f : \{0,1\}^n \to \{0,1\}^n$ in a point $v \in \{0,1\}^n$

$$\partial_j f_i(v) := \frac{f_i(v^{\{j\}}) - f_i(v)}{v_j^{\{j\}} - v_j}. \tag{1.3.1}$$

Furthermore, we denote with ∇f_i the vector $\left(\partial_1 f_i \quad , \ldots, \quad \partial_n f_i\right)^t$.

Using the discrete derivative we construct a so-called *interaction graph* corresponding to the function f consisting of n nodes and signed edges between them. This graph can be seen as a discrete version of the Jacobian matrix. We furthermore distinguish between a local version of this graph representing the influence of the components on each other in neighborhood of a specific state and a global version capturing all possible influences.

Definition 1.28. The *local interaction graph* $IG_f(x) := (V, E)$, $x \in \{0,1\}^n$ of a Boolean function $f : \{0,1\}^n \to \{0,1\}^n$ consists of n nodes $V := [n]$ consisting of the components of the BN, and a signed edge-set $E\big(IG_f(x)\big) \subseteq V \times V \times \{-1, 1\}$, which is defined by

$$(i, j, \epsilon) \in E\big(IG_f(x)\big) :\Leftrightarrow (\partial_i f_j)(x) = \epsilon.$$

We denote with $IG^{\text{global}}(f)$ – the *global interaction graph* – the union of all local interaction graphs. I.e.

$$IG^{\text{global}}(f) := \bigcup_{x \in \{0,1\}^n} IG_f(x).$$

We remark that in the local version of the interaction graph there can be maximally one edge between two components. However, in the global version it can happen that two edges with opposite sign exist between two components. This means that a component can act activating and inhibiting on another component depending on the values of the other components of the network. However, for the majority of BNs modelings GRNs the global interaction graph has not more than one edge between different components. We will therefore often make this assumption. Edges in the interaction graph with positive label we also call *activating edges*, while edges with negative label are also called *inhibitory edges*.

We conclude this section with a small example.

Example 1.2. We continue Example 1.1. In Figure 1.3.3 its global interaction graph is depicted. To check for example whether there is an edge with negative label from the second to the third component consider the state 0000 and the discrete derivative $\partial_2 f_3$, i.e.

$$\partial_2 f_3(0000) = \frac{f_3(0100) - f_3(0000)}{1 - 0} = \frac{0 - 1}{1 - 0} = -1.$$

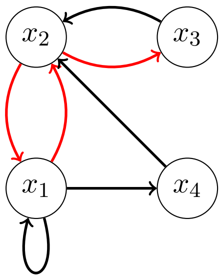


FIGURE 1.3.3. $IG^{\text{global}}(f)$ from Example 1.2. Red edges denote edges with negative label. Black edges denote edges with positive label.

CHAPTER 2

Conversion of Boolean models into ODE models

Contents

2.1. The transformation algorithm explained **24**
2.1.1. Multivariate polynomial interpolation 24
2.1.2. Concatenation with Hill functions 25
2.1.3. Simplifying assumptions made in this chapter 30
2.2. Preservation of steady states and trap spaces **31**
2.2.1. Preservation of steady states 31
2.2.2. Preservation of trap spaces 32
2.3. Application: a small example **35**
2.4. Correspondence of trap spaces in detail **36**
2.4.1. Correspondence of trap spaces for normalized Hill functions 36
2.4.2. Associating invariant sets to trap spaces 38
2.4.2.1. Invariance in the time-discrete dynamical system 39
2.4.2.2. Invariance in the ODE system 40
2.4.3. Computing explicit values for the Hill coefficients to guarantee invariance 42
2.5. Discussion **44**

In this chapter we focus on two hybrid approaches transforming Boolean networks into ODE systems. The value of such hybrid approaches crucially depends on our ability to transfer knowledge from the Boolean model to the continuous model. Results available so far are limited. They concern mainly static properties such as steady states Wittmann et al. [2009], Veliz-Cuba et al. [2012] and trap spaces Schwieger et al. [2018]. The algorithm considered here stems from Wittmann et al. [2009]. Already in Wittmann et al. [2009] and later more generalized in Veliz-Cuba et al. [2012] it was proven that information about the location and number of steady states can be deduced from the Boolean model. Less is known about other dynamical features such as oscillations. However, there are some results for slightly different hybrid approaches using piecewise-linear differential equations about oscillatory behavior, for example in Snoussi [1989], Veliz-Cuba et al. [2014b].

The focus of this chapter lies on the presentation of results in Schwieger et al. [2018] concerning trap spaces (see Definition 1.27). Trap spaces are a characteristic feature of dynamical systems that generalize the concept of steady states - namely subspaces of the state space where only some of the components remain fixed. Trap spaces have been frequently studied in the past Zañudo and Albert [2015, 2013], Remy and Ruet [2008], Siebert [2009] in the context of Boolean networks. The goal of this chapter is to show that these trap spaces of a Boolean dynamical system correspond to invariant sets of the continuous dynamical system, whereby the size of these sets can be controlled by the Hill exponents. Trap spaces in a Boolean network can be computed efficiently (see Klarner et al. [2015]). This hints to possible model reductions whereby only a specific subnetwork of the original Boolean network needs to be translated into an ODE model.

Chapter 2 is organized as follows. In Section 2.1 we briefly review the transformation algorithm from Wittmann et al. [2009], which assigns to a Boolean function a parametrized family of ODE systems. In Section 2.2 it is shown that trap spaces of a Boolean network cannot directly be linked to trap spaces in the ODE model in the general case. This motivates the rest of the chapter. We start by considering the case of normalized Hill functions in Section 2.4.1. In Section 2.4.2, it is shown that, even if the Hill functions are not normalized, a correspondence to invariant sets is possible. The size of these sets can be controlled by the parameters of the Hill functions, as we will demonstrate in Section 2.4.3.

2.1. The transformation algorithm explained

Different methods have been proposed to transfer either Boolean networks to ODE models in an automated framework (see, e.g., Wittmann et al. [2009], Krumsiek et al. [2010]) or vice versa (see, e.g., Stötzel et al. [2015], Glass and Kauffman [1973]). The most common way of modeling is to first construct a Boolean network and to derive an ODE system in the second step. In this chapter we concentrate on the algorithm presented in Wittmann et al. [2009]. This algorithm aims at preserving the network topology and the type of influence (activating or inhibiting) during the conversion process by using multivariate polynomial interpolation and Hill functions.

Consider an arbitrary Boolean function $f : \{0,1\}^n \to \{0,1\}^n$, $n \in \mathbb{N}$. The function f could induce a state transition graph via an asynchronous or a synchronous update-rule. But in this chapter it is enough to consider only the synchronous update. Since we will investigate trap spaces and steady states, which stay the same in both kinds of BNs, we can solely focus on synchronous BNs. Hence, we consider a synchronous Boolean network (see Definition 1.22), which we express in this chapter in the following way:

$$\begin{aligned} x^{t+1} &= f(x^t), t \in \mathbb{N}, \\ x^0 &:= x_0 \in \{0,1\}^n. \end{aligned} \tag{2.1.1}$$

The Boolean function f is in two steps converted into a family of ODE systems:

2.1.1. Multivariate polynomial interpolation. First, a multivariate polynomial interpolation of f is constructed to obtain a continuous continuation of f.

Definition 2.1. Let $f : \{0,1\}^n \to \{0,1\}^n$ be a Boolean function. Its *multivariate polynomial interpolation* is a function $\overline{I}(f) : [0,1]^n \to [0,1]^n$ with $\overline{I}(f)_i := \overline{I}(f_i) : [0,1]^n \to [0,1]$ that assigns polynomials to the Boolean functions f_i given explicitly by

$$\overline{I}(f_i)(\overline{x}_1, \dots, \overline{x}_n) = \sum_{x \in \{0,1\}^n} f_i(x) \prod_{j=1}^{n} \big(x_j \overline{x}_j + (1 - x_j)(1 - \overline{x}_j)\big). \tag{2.1.2}$$

x_1	x_2	x_3	$f_1(x)$	$f_2(x)$	$f_3(x)$
0	0	0	1	0	1
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	1	0	0
1	1	1	1	1	0

Table 2.1.1. Example 2.1.

As shown in [Wittmann et al., 2009, p.16-17] the multivariate polynomial interpolation $\overline{I}(f_i)$ is the unique polynomial of minimal degree[1] that coincides with f_i on the vertices $x \in \{0,1\}^n$ of the hypercube.

We illustrate this process with a small example which will be used again later on:

Example 2.1. Consider the Boolean function $f : \{0,1\}^3 \to \{0,1\}^3$ defined in Table 2.1.1. Using (2.1.2) we obtain the polynomial map $\overline{I}(f)$:

$$\overline{I}(f) = \begin{pmatrix} (1-\overline{x}_1)(1-\overline{x}_2)(1-\overline{x}_3) + \overline{x}_1(1-\overline{x}_2)\overline{x}_3 + \overline{x}_1\overline{x}_2\overline{x}_3 + \overline{x}_1\overline{x}_2(1-\overline{x}_3) + \overline{x}_1(1-\overline{x}_2)(1-\overline{x}_3) \\ \overline{x}_1(1-\overline{x}_2)\overline{x}_3 + \overline{x}_1\overline{x}_2\overline{x}_3 + (1-\overline{x}_1)(1-\overline{x}_2)\overline{x}_3 + (1-\overline{x}_1)\overline{x}_2\overline{x}_3 \\ (1-\overline{x}_1)(1-\overline{x}_2)(1-\overline{x}_3) + \overline{x}_1(1-\overline{x}_2)\overline{x}_3 + \overline{x}_1(1-\overline{x}_2)(1-\overline{x}_3) + (1-\overline{x}_1)(1-\overline{x}_2)\overline{x}_3 \end{pmatrix}$$

$$= \begin{pmatrix} (1-\overline{x}_1)(1-(\overline{x}_2+\overline{x}_3-\overline{x}_2\overline{x}_3)) + \overline{x}_1 \\ \overline{x}_3 \\ 1-\overline{x}_2 \end{pmatrix}$$

2.1.2. Concatenation with Hill functions.

We could use already the function $\overline{I}(f)$ to generate an ODE-system. However, experiments suggest that in many GRNs the components obey a switch-like behavior Krumsiek et al. [2010]. In Section 1.2.2 we gave a motivation for the use of these functions. Therefore, in a next step, the function $\overline{I}(f)$ is concatenated with Hill functions. The exact shape of these Hill functions can be controlled by specific parameters. This allows to use the parameters of the Hill functions to induce a behavior of the resulting continuous function similar to step functions.

Definition 2.2. A multivariate *Hill function* is defined by

$$H^{\vec{k},\vec{\theta}} : [0,1]^n \to [0,1]^n,$$
$$H^{\vec{k},\vec{\theta}}(x) = \left(h_{k_1,\theta_1}(x_1), \quad \ldots \quad , h_{k_n,\theta_n}(x_n)\right)$$

with univariate Hill functions

$$h_{k,\theta} : [0,1] \to [0,1],$$
$$h_{k,\theta}(x) = \frac{x^k}{x^k + \theta^k}, \qquad k \in \mathbb{R}_{>0}, \theta \in (0,1).$$

The parameter k is called *Hill coefficient* and θ is called *threshold* (see also Section 1.2.2). In Figure 2.1.1 the graphs of several Hill functions with

[1] The degree of a polynomial $P(X_1,\ldots,X_n) = \sum_{(m_1,\ldots,m_n)\in\mathbb{N}^n} a_{m_1,\ldots,m_n} X_1^{m_1} \cdot \ldots \cdot X_n^{m_n}$ is defined as $\deg(P) = \max_{(m_1,\ldots,m_n)\in\mathbb{N}^n, a_{m_1,\ldots,m_n}\neq 0} \sum_{i=1}^{n} m_i$.

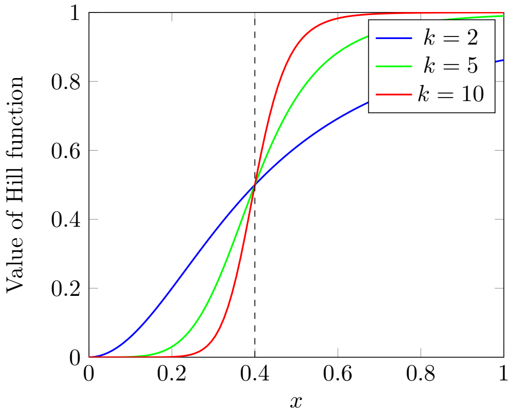


FIGURE 2.1.1. Different Hill functions for $\theta = 0.4$.

different Hill coefficients are depicted. A larger Hill coefficient leads to a steeper Hill function. By construction it holds $h_{k,\theta}(0) = 0$ and $h_{k,\theta}(1) < 1$ for any $\theta \in (0,1)$, $k \in \mathbb{R}_{>0}$.

Now the continuous variables in the function $\overline{I}(f)$ are replaced by Hill functions[2] with distinct Hill coefficients and thresholds for each variable and each component they influence. In this way we obtain a continuous function $\overline{f} : [0,1]^n \to [0,1]^n$ with the Hill coefficients and thresholds as parameters.[3] The function $\overline{f}$ should be understood as a continuous continuation of f on $[0,1]^n$. This means that $\overline{f}$ agrees with f on $\{0,1\}^n$ or, vaguely speaking, takes at least similar values on $\{0,1\}^n$.

Finally, in a last step *life-times* and *degradation terms* are added:

$$\begin{aligned} \dot{\overline{x}} &= D \cdot (\overline{f}(\overline{x}) - \overline{x}), \\ \overline{x}(0) &= \overline{x}_0 \in [0,1]^n, \end{aligned} \tag{2.1.3}$$

where D is a strictly positive diagonal matrix, $D = \mathrm{diag}(d_1, \dots, d_n)$ with $d_i > 0$ for $i \in [n]$[4]. A solution $\overline{x} : \mathbb{R}_{\geq 0} \to [0,1]^n$ of (2.1.3) could for example represent the course of mRNA or protein concentrations over time. Throughout this chapter, the line over variables and functions is used only in the continuous setting.

Let us now consider a few examples to get familiar with this conversion algorithm. We will use these examples also to get a first feeling for the resemblance of ODE models and Boolean models of the same regulatory network.

[2] Sometimes it is an advantage to consider normalized Hill functions $\frac{h_{k,\theta}(\overline{x})}{h_{k,\theta}(1)}$ instead of Hill functions in the above procedure. We refer for details to Wittmann et al. [2009] and the next section.

[3] I.e. $\overline{f}_1 := \overline{I}(f_1) \circ H^{\vec{k}^1,\vec{\theta}^1}, \dots, \overline{f}_n := \overline{I}(f_n) \circ H^{\vec{k^n},\vec{\theta^n}}$ and $\vec{k}^i \in \mathbb{R}^n_{>0}$, $\vec{\theta^i} \in (0,1)^n$, $i \in [n]$.

[4] We could also consider the ODE system $\dot{x} = \mathrm{diag}(\vec{\alpha})\overline{f}^{\vec{k}}(x) - \mathrm{diag}(\vec{\beta})x$. However, after a normalization step we arrive at a system of the form (2.1.4) (see Wittmann et al. [2009]).

Example 2.2. Consider the Boolean function $f(x) = \begin{pmatrix} x_1 \vee \neg x_2 \\ \neg x_1 \vee x_2 \end{pmatrix}$. Using the above procedure we obtain the continuous function $\overline{f}$ given by

$$\overline{f}_1(\overline{x}_1, \overline{x}_2) = \big[h_{k_{11},\theta_{11}}(\overline{x}_1) + (1 - h_{k_{12},\theta_{12}}(\overline{x}_2)) - h_{k_{11},\theta_{11}}(\overline{x}_1) \cdot (1 - h_{k_{12},\theta_{12}}(\overline{x}_2))\big],$$
$$\overline{f}_2(\overline{x}_1, \overline{x}_2) = \big[(1 - h_{k_{21},\theta_{21}}(\overline{x}_1)) + h_{k_{22},\theta_{22}}(\overline{x}_2) - (1 - h_{k_{21},\theta_{21}}(\overline{x}_1)) \cdot h_{k_{22},\theta_{22}}(\overline{x}_2)\big].$$

This leads us to the family of ODE-system

$$\dot{\overline{x}}_1 = d_1 \cdot (\overline{f}_1(\overline{x}_1, \overline{x}_2) - \overline{x}_1),$$
$$\dot{\overline{x}}_2 = d_2 \cdot (\overline{f}_2(\overline{x}_1, \overline{x}_2) - \overline{x}_2).$$

Choosing different values for $\Theta = \begin{pmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{pmatrix}$, $k = \begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix}$ and $d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$ we obtain specific ODE systems. Let us take a look at a few different trajectories obtained by using random parametrizations[5] starting in the point $(0, 0)$ (see Figure 2.1.2). The Boolean function f has three steady states, namely 10, 01, 11. As we will see later (Theorem 2.1) these steady states are preserved under the conversion into an ODE-system if the Hill coefficients are chosen sufficiently large. Taking a look at the three different trajectories depicted in Figure 2.1.2, we see that indeed there are parametrizations for which the trajectory starting in $(0, 0)$ converges to each of the three fixed points.

In the first case[6], depicted in Figure 2.1.2a the trajectory converges towards a fixed point close to $(1, 1)$. In Figure 2.1.2b we see a trajectory converging towards a fixed point close to $(1, 0)$.[7] Finally, in Figure 2.1.2c we see a trajectory converging in a fixed point close to $(0, 1)$.[8] In all three cases we see that the trajectory is almost linear as long as the concentrations are below the thresholds. Then approaching the thresholds the trajectories rapidly change their direction.

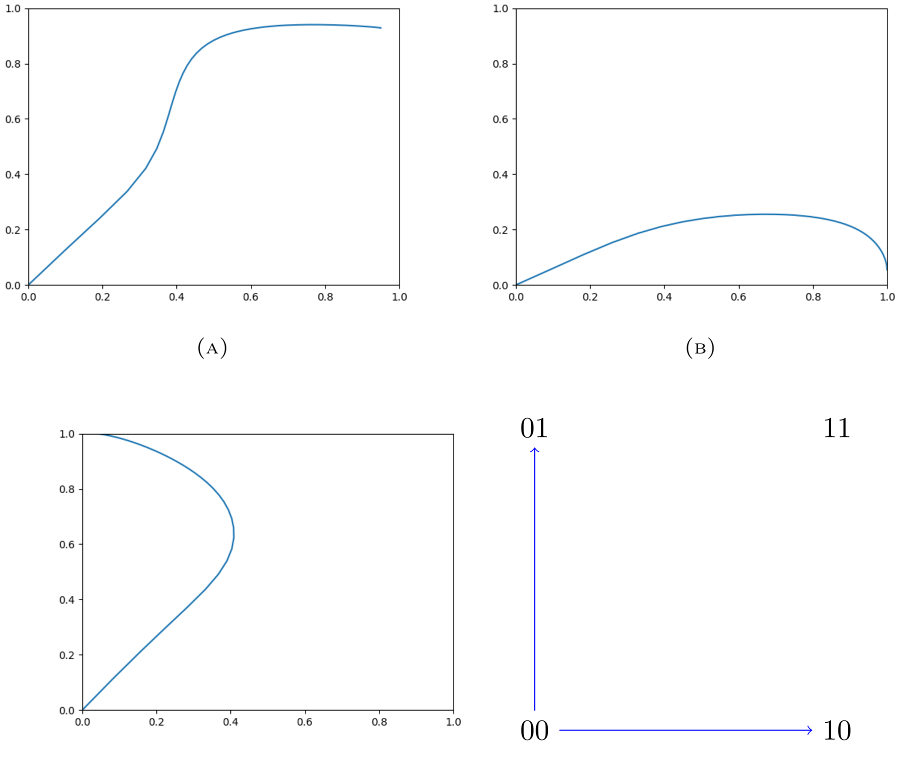
The fact that the trajectories starting in the zero point converge in our numerical analysis always towards one of the three fixed points allows us to compare the trajectories of the ODE-system with the trajectories in the graph $G_{\text{async}}(f)$ and $G_{\text{sync}}(f)$. In the case of a synchronous update only the state 11 is reachable from 00. However, the comparison with the ODE-system suggests that we should be cautious with this conclusion. Often it is argued that the asynchronous update, leads to more realistic conclusions.

[5] Thresholds were chosen uniformly from the interval $[0.4, 0.6]$, Hill coefficients from the interval $[3, 6]$ and lifetimes from $[1, 5]$.

[6] with parameters rounded to the second digit $\Theta = \begin{pmatrix} 0.4 & 0.44 \\ 0.58 & 0.50 \end{pmatrix}$ and $k = \begin{pmatrix} 3.59 & 5.96 \\ 5.46 & 4.06 \end{pmatrix}$ and $d = \begin{pmatrix} 3.57 \\ 4.59 \end{pmatrix}$

[7] with parameters rounded to the second digit $\Theta = \begin{pmatrix} 0.50 & 0.54 \\ 0.41 & 0.42 \end{pmatrix}$ and $k = \begin{pmatrix} 5.69 & 5.0 \\ 3.18 & 4.34 \end{pmatrix}$ and $d = \begin{pmatrix} 3.31 \\ 1.95 \end{pmatrix}$

[8] with parameters rounded to the second digit $\Theta = \begin{pmatrix} 0.58 & 0.56 \\ 0.47 & 0.41 \end{pmatrix}$ and $k = \begin{pmatrix} 5.56 & 5.75 \\ 4.79 & 4.06 \end{pmatrix}$ and $d = \begin{pmatrix} 2.8 \\ 4.0 \end{pmatrix}$



(A)

(B)

(C)

(D) $G_{\text{async}}(f)$.

FIGURE 2.1.2. Illustration of Example 2.2: on the horizontal axis is the concentration of x_1 depicted and on the vertical axis the concentration of x_2. We see trajectories starting in $(0, 0)$ of the ODE-system for different parametrizations and the ASTG of the Boolean function f in Example 2.2.

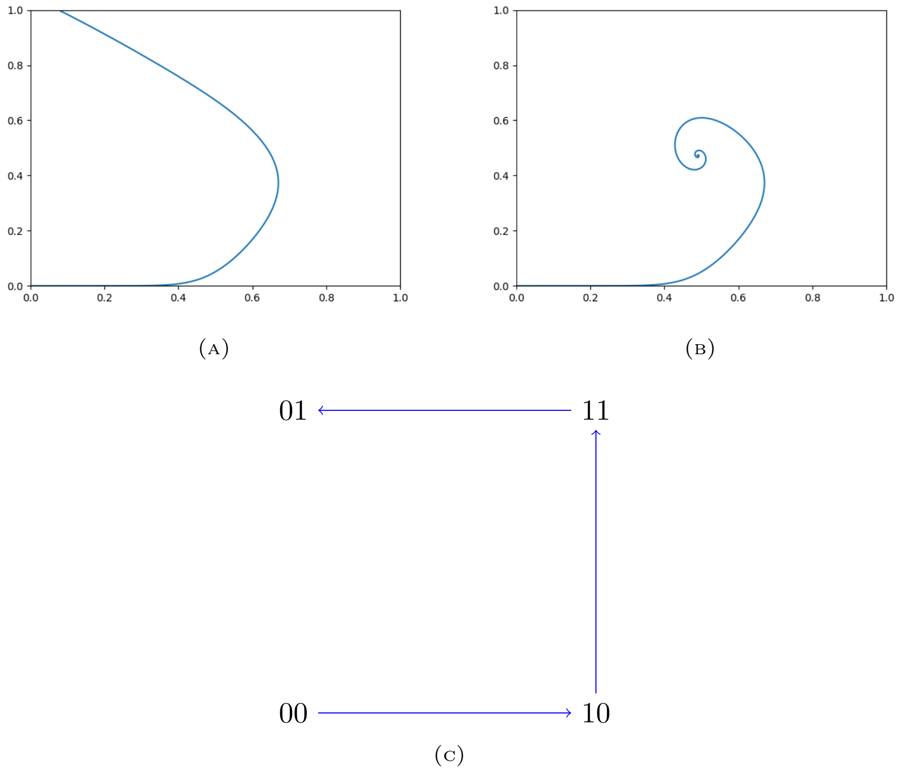
The graph $G_{\text{async}}(f)$, depicted in Figure 2.1.2d, shows, that from the state 00 we can reach the steady states 10 and 01, but not the steady state 11. However, Figure 2.1.2a shows us, that the case where both components of the Boolean network are updated simultaneously cannot be neglected always. Consequently, we should be very cautious about conclusions drawn from reachability properties of the $G_{\text{async}}(f)$ as well.

Let us look at another example:

Example 2.3. Consider the Boolean function

$$f(x) = \begin{pmatrix} \neg x_2 \\ x_1 \vee x_2 \end{pmatrix}.$$

In Figure 2.1.3c we see the ASTG of f. The ASTG is characterized by a long trajectory starting in the state 00 and ending in the state 01. Let us look at different parametrizations of the resulting ODE system. In Figure 2.1.3a



(A)

(B)

(C)

Figure 2.1.3. Illustration of Example 2.3: on the horizontal axis is the concentration of x_1 depicted and on the vertical axis the concentration of x_2. We see trajectories starting in $(0,0)$ of the ODE-system for different parametrizations and the ASTG of the Boolean function f in Example 2.3.

we see a trajectory[9] qualitatively resembling the discrete trajectory[10] and ending in a fixed point which can be considered a continuous homologue of the discrete steady state 01.

In Figure 2.1.3b we see a trajectory[11] spiraling inward towards a fixed point. This is a behavior not predicted in the Boolean model, since firstly the fixed point in the middle has no discrete counterpart. Secondly in $G_{\text{async}}(f)$ we do not find a transition 01 → 00 which could explain the spiraling effect.

The above examples show that in principle the obtained ODE-system can behave very differently from the Boolean network it was stemming from.

[9] with parameters rounded to the second digit $\Theta = \begin{pmatrix} 0 & 0.47 \\ 0.5 & 0.51 \end{pmatrix}$ and $k = \begin{pmatrix} 0 & 3.2 \\ 8.7 & 8.27 \end{pmatrix}$ and $d = \begin{pmatrix} 2.44 \\ 2.13 \end{pmatrix}$

[10] i.e. the walk in the state transition graph of the Boolean network

[11] with parameters rounded to the second digit $\Theta = \begin{pmatrix} 0 & 0.47 \\ 0.5 & 0.7 \end{pmatrix}$ and $k = \begin{pmatrix} 0 & 3.12 \\ 8.66 & 8.26 \end{pmatrix}$ and $d = \begin{pmatrix} 2.44 \\ 2.14 \end{pmatrix}$

More extended studies on example systems show that many dynamical features, in general, are not necessarily preserved in Hill-type continuous models (see e.g. Saadatpour and Albert [2016], Chaves and Preto [2013], Ackermann et al. [2012], Gehrmann and Drossel [2010]). We will focus now on a few aspects of the dynamics where similarities can be proven in a systematic way.

2.1.3. Simplifying assumptions made in this chapter. In this chapter we will mostly consider a specific case of the above methodology. We will attribute to each variable in $\overline{I}(f)$ a fixed Hill function independently in which component function the variable appears. This allows us to associate to f a family of two-parametric continuous functions $(\overline{f}^{\vec{k},\vec{\theta}})_{\vec{k}\in\mathbb{R}^n_{>0},\vec{\theta}\in(0,1)^n}$, where $\overline{f}^{\vec{k},\vec{\theta}} : [0,1]^n \to [0,1]^n$. The construction of $\overline{f}^{\vec{k},\vec{\theta}}$ will be explained in a moment. This leads to a time-discrete but state-continuous dynamical system of the form:

$$\overline{x}^{t+1} = \overline{f}^{\vec{k},\vec{\theta}}(\overline{x}^t), t \in \mathbb{N},$$
$$\overline{x}^0 := \overline{x}_0 \in [0,1]^n.$$

Then, as in the general case, we construct for any $\vec{k} \in \mathbb{R}^n_{>0}$ and $\vec{\theta} \in (0,1)^n$ in a second step an ODE system of the form:

$$\dot{\overline{x}} = D \cdot (\overline{f}^{\vec{k},\vec{\theta}}(\overline{x}) - \overline{x}), \qquad (2.1.4)$$
$$\overline{x}(0) = \overline{x}_0 \in [0,1]^n,$$

where D is as before a strictly positive diagonal matrix, $D = \mathrm{diag}(d_1, \dots, d_n)$ with $d_i > 0$ for $i \in [n]$[12].

Now we turn our attention to the construction of the functions $\overline{f}^{\vec{k},\vec{\theta}}$. According to Wittmann et al. [2009], we will analyze here two ways to construct the family $\overline{f}^{\vec{k},\vec{\theta}}$, $\vec{k} \in \mathbb{R}^n_{>0}, \vec{\theta} \in (0,1)^n$, one based on Hill cubes and the other one based on normalized Hill cubes. The basic procedure in both approaches is the same.

However, in Wittmann et al. [2009] the Hill-coefficients $\vec{k}$ and the thresholds $\vec{\theta}$ are allowed to differ in each of the components of the functions and variables. Here, in contrast we will assume that they can only differ between different variables. This allows us to represent $\overline{f}^{\vec{k},\vec{\theta}}$ as a concatenation of two functions.

Definition 2.3. A *Hill cube* is defined as ([Wittmann et al., 2009, p. 5]):

$$\overline{f}^{\vec{k},\vec{\theta}} : [0,1]^n \to [0,1]^n,$$
$$\overline{f}^{\vec{k},\vec{\theta}} := \overline{I}(f) \circ H^{\vec{k},\vec{\theta}}.$$

[12] We could also consider the ODE system $\dot{x} = \mathrm{diag}(\vec{\alpha})\overline{f}^{\vec{k}}(x) - \mathrm{diag}(\vec{\beta})x$. However, after a normalization step we arrive at a system of the form (2.1.4) (see Wittmann et al. [2009]).

A *normalized Hill cube* is defined as ([Wittmann et al., 2009, p. 5]):

$$\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}} : [0,1]^n \to [0,1]^n,$$

$$\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}} := \overline{I}(f) \circ \frac{H^{\vec{k},\vec{\theta}}}{H^{\vec{k},\vec{\theta}}(\vec{1})},$$

where the division $\frac{H^{\vec{k},\vec{\theta}}}{H^{\vec{k},\vec{\theta}}(\vec{1})}$ is meant component wise.

Normalized Hill cubes have the advantage that the continuous function $\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}$ obtained by the conversion takes the same values as f on the vertices of the hypercube $\{0,1\}^n$. However, this perfect agreement is not a plausible biological assumption, since biological interactions such as enzyme kinetics are known to only asymptotically approach a saturation level [Wittmann et al., 2009, p. 15].

2.2. Preservation of steady states and trap spaces

The normalization of the Hill cubes implies that $\overline{f}^{\vec{k},\vec{\theta}}$ coincides with f on the vertices of the hypercube $[0,1]^n$. If the Hill cubes are not normalized, the function $\overline{f}^{\vec{k},\vec{\theta}}$ will differ slightly from f on $\{0,1\}^n$. This raises the question which dynamical properties are preserved during the conversion from the state and time discrete model to the continuous model. Here we will consider steady states and trap spaces.

2.2.1. Preservation of steady states. A fixed point $\overline{x} \in [0,1]^n$ of the ODE system is defined as the zero locus of (2.1.4),

$$\vec{0} = \overline{f}^{\vec{k},\vec{\theta}}(\overline{x}) - \overline{x}.$$

In the context of (asynchronous or synchronous) Boolean networks such fixed points are called steady states.[13] For sufficiently large $\vec{k}$ we can find a continuous equivalent of a steady state $x_{\text{steady}} \in \{0,1\}^n$ of the Boolean function f in a neighborhood $U(x_{\text{steady}})$ of x_{steady}. This result was derived in Wittmann et al. [2009] and generalized in Veliz-Cuba et al. [2012].

Theorem 2.1 (Wittmann et al. [2009])**.** Assume $x_{\text{steady}} \in \{0,1\}^n$ is a steady state of $f \in \mathbb{B}(n,n)$. Consider the ODE model (2.1.4). If $\min_{i\in[n]} k_i$ is sufficiently large, then there is a neighborhood $U(x_{\text{steady}}) \subseteq [0,1]^n$ of x_{steady} such that the ODE model (2.1.4) has a steady state $\overline{x}^{\vec{k}}_{\text{steady}} \in U(x_{\text{steady}})$. Moreover, for any sequence $(\vec{k}^j)_{j\in\mathbb{N}} \subset \mathbb{R}^n_{>0}$ that converges component-wise to infinity, it holds

$$\lim_{j\to\infty} \overline{x}^{\vec{k}^j}_{\text{steady}} = x_{\text{steady}}.$$

[13] Note that for synchronous and asynchronous BNs this agrees with Definition 1.9, since the states of the STG induced by the update function $f \in \mathbb{B}(n,n)$ of the BN that have no outgoing transitions are precisely the states for which $f(x) = x$ holds.

2.2.2. Preservation of trap spaces.

Trap spaces represent a generalization of steady states or fixed points in that only a few components remain unchanged, whereas in steady states all components remain fixed. To simplify matters we will restrict ourselves to the case where the first $m \leq n$ components stay fixed (see also Definition 1.27). However, the results remain of course true in the general case because we can always permute the components of f in such a way that we arrive at this special case.

Definition 2.4. Consider a dynamical system (G, X^n, ϕ) with time domain $G = \mathbb{N}$ or $G = \mathbb{R}_{\geq 0}$, $n \in \mathbb{N}$, a non-empty set X, and evolution ϕ given by

$$\begin{aligned} \phi : G \times X^n &\to X^n, \\ (t, x) &\mapsto \phi_t(x). \end{aligned}$$

We define the set

$$(p, *)_X = \{x \in X^n \big| x_i = p_i \text{ for } i \leq m\}$$

with $p \in X^m$, $m \in [n]$. We call a set of the form $(p, *)_X$ a *trap space* of the dynamical system if it is invariant with respect to the evolution, i.e.

$$\phi_t(x) \in (p, *)_X \quad \forall x \in (p, *)_X,\, t \in G.$$

If the meaning of X is clear from the context, we write $(p, *)$ instead of $(p, *)_X$.

If we choose in the above definition $m = n$ then we arrive at the definition of a fixed point. Therefore, every fixed point is as well a trap space. On the other hand we excluded the trivial case where no component is fixed from the definition.

Remark 2.1. Note that Definition 2.4 agrees with Definition 1.27 for synchronous BNs. However, in the context of this chapter the formulation in Definition 2.4 allows for a more convenient comparison between the different dynamical systems considered here.

As told before the normalization of the Hill cubes implies that $\overline{f}^{\vec{k},\vec{\theta}}$ coincides with f on the vertices of the hypercube $[0,1]^n$. If the Hill cubes are not normalized, the function $\overline{f}^{\vec{k},\vec{\theta}}$ will differ slightly from f on $\{0,1\}^n$. Nevertheless, as we saw before in Theorem 2.1 the steady states are still preserved for suitable parametrizations. This motivates the question if $\overline{f}^{\vec{k},\vec{\theta}}$ and $\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}$ inherit the trap spaces of f as well? A natural idea would be to generalize Theorem 2.1 by assigning to each trap space of f a trap space of $\overline{f}^{\vec{k},\vec{\theta}}$. However, as the following example shows, this does not always work.

Example 2.4. Consider again the Boolean function $f : \{0,1\}^3 \to \{0,1\}^3$ from Example 2.1. We can write $\overline{I}(f)$ in the form

$$\overline{I}(f)(\overline{x}) = \begin{pmatrix} 1 - (1 - \overline{x}_1) \cdot \underbrace{(\overline{x}_2 + \overline{x}_3 - \overline{x}_2\overline{x}_3)}_{=:C(\overline{x}_2,\overline{x}_3)} \\ \overline{x}_3 \\ 1 - \overline{x}_2 \end{pmatrix}.$$

From the definition of f (see Table 2.1.1 it is clear that $(1,*) \subseteq \{0,1\}^3$ is a trap space of f. We want to know if there is a $\overline{p}_1 \in [0,1]$ such that $(\overline{p}_1,*) \subseteq [0,1]^3$ is a trap space of $\overline{f}^{\vec{k},\vec{\theta}}$. For simplicity, we assume $\vec{k} = \begin{pmatrix} k & k & k \end{pmatrix}^T$ for some $k \in \mathbb{R}_{>0}$, $\vec{\theta} = \begin{pmatrix} \theta & \theta & \theta \end{pmatrix}^T$, $\theta \in (0,1)$, and consequently write $\overline{f}^{k,\theta} = \overline{f}^{\vec{k},\vec{\theta}}$. This means, we need to find $\overline{x}_1 \in [0,1]$ such that

$$\forall \overline{x}_2, \overline{x}_3 \in [0,1] : 0 = \overline{f}_1^k(\overline{x}_1, \overline{x}_2, \overline{x}_3) - \overline{x}_1$$

holds. We have

$$\overline{f}_1^k(\overline{x}_1, \overline{x}_2, \overline{x}_3) - \overline{x}_1 = \overline{I}(f_1) \circ H^{k,\theta}(\overline{x}_1, \overline{x}_2, \overline{x}_3) - \overline{x}_1.$$

In order to find a trap space of $\overline{f}^{k,\theta}$, we can instead look at the function $g : [0,1]^3 \to [0,1]$

$$\overline{g}(\overline{x}_1, \overline{y}_1, \overline{y}_2) := \overline{I}(f_1)(h_{k,\theta}(\overline{x}_1), \overline{y}_1, \overline{y}_2) - \overline{x}_1,$$

since the input variables of the Hill functions are allowed to vary freely. Consequently, for a fixed $\overline{x}_1 \in [0,1]$, we can interpret the function $g(\overline{x}_1, \cdot, \cdot)$ as a polynomial $\overline{g}(\overline{x}_1, Y_1, Y_2)$ in $\mathbb{R}[Y_1, Y_2]$. Since the set $K := h_{k,\theta}([0,1]) \times h_{k,\theta}([0,1])$ has a nonempty interior, we have the following equivalence:

$$\begin{aligned} &\exists \overline{x}_1 \in [0,1] : 0 = \overline{g}(\overline{x}_1, Y_1, Y_2) \in \mathbb{R}[Y_1, Y_2] \\ \Leftrightarrow\ &\exists \overline{x}_1 \in [0,1] \forall (\overline{y}_1, \overline{y}_2) \in K : 0 = \overline{g}(\overline{x}_1, \overline{y}_1, \overline{y}_2) \\ \Leftrightarrow\ &\exists \overline{x}_1 \in [0,1] \forall \overline{x}_2, \overline{x}_3 \in [0,1] : 0 = \overline{f}_1^k(\overline{x}_1, \overline{x}_2, \overline{x}_3) - \overline{x}_1 \end{aligned}$$

Hence, for $(\overline{x}_1, *)$ being a trap space, the polynomial system of equations needs to be fulfilled:

$$\begin{aligned} &0 = \overline{g}(\overline{x}_1, Y_1, Y_2) \\ \Leftrightarrow\ &0 = 1 - (1 - \frac{\overline{x}_1^k}{\theta^k + \overline{x}_1^k}) \cdot C(Y_1, Y_2) - \overline{x}_1 \end{aligned} \tag{2.2.1}$$

Since the polynomial $C(Y_1, Y_2)$ is not the zero polynomial and it has no constant monomials, we obtain the system of equations

$$\begin{aligned} 1 - \frac{\overline{x}_1^k}{\theta^k + \overline{x}_1^k} &= 0, \\ 1 - \overline{x}_1 &= 0, \end{aligned}$$

which has no solution for $k > 0$, $\theta \in (0,1)$. This means there is no $k \in \mathbb{R}_{>0}$ such that $\overline{f}^{k,\theta}$ has a trap space of the form $(\overline{x}_1, *)$, $\overline{x}_1 \in [0,1]$.

This example shows that we cannot link the trap spaces of a Boolean system directly to the trap spaces of the ODE system if we use Hill functions. That means, we cannot generalize Theorem 2.1 directly. One reason why these difficulties arise at all lies in the fact that the functions $\overline{f}^{\vec{k},\vec{\theta}}$ do not agree perfectly with their Boolean counterparts on $\{0,1\}^n$. Instead of associating the trap spaces of f to trap spaces of $\overline{f}^{\vec{k},\vec{\theta}}$ or $\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}$, we can associate them to certain invariant sets of these functions, whose size can be controlled by the Hill coefficients $\vec{k}$. In contrast to trap spaces the previously fixed

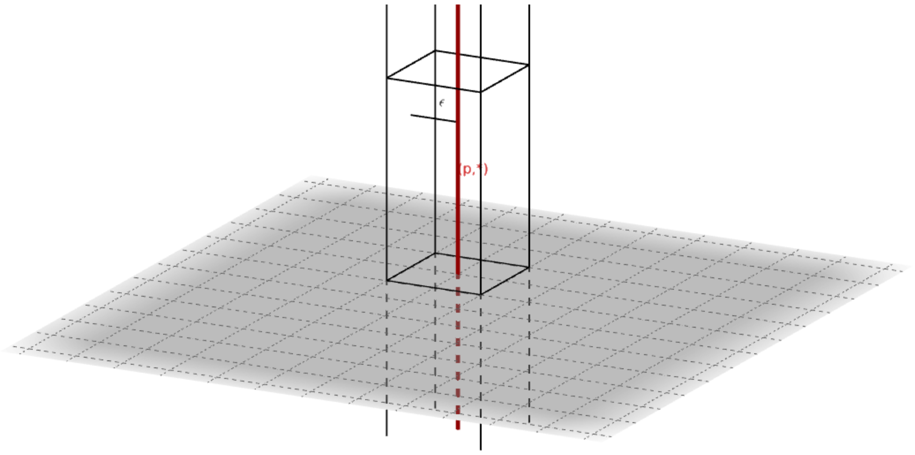


FIGURE 2.2.1. Illustration of the set $K(\overline{p}, \epsilon)$.

components are allowed to change to a certain degree over time in these invariant sets. We will now state what we mean with this precisely.

Definition 2.5. Let $\text{proj}^{[m]} : [0,1]^n \to [0,1]^m$ be the projection on the first $m \leq n$ components. For any $\epsilon > 0$ and $(\overline{p}, *) \subseteq [0,1]^n$, $\overline{p} \in [0,1]^m$ we define (see Figure 2.2.1)

$$K(\overline{p}, \epsilon) := \left\{\overline{x} \in [0,1]^n \middle| \|\text{proj}^{[m]}(\overline{x}) - \overline{p}\|_\infty \leq \epsilon\right\}.$$

Then it it is possible to obtain the following result if we use the construction explained in Section 2.1.3.

Proposition 2.1 (Schwieger et al. [2018]). Let $(\overline{p}, *)$, $\overline{p} \in \{0,1\}^m$ be a trap space of $f \in \mathbb{B}(n,n)$. Then

$$\forall \epsilon \in (0, \theta_{\text{min}}) \exists \vec{k_0} \in \mathbb{N}^n \forall \vec{k} \geq \vec{k_0} \forall x_0 \in K(\overline{p}, \epsilon) : \forall t \in [0, \infty) : x(t) \in K(\overline{p}, \epsilon),$$

where $x(t)$ is the solution of (2.1.4) with initial condition $x(0) = x_0$ and $\theta_{\text{min}} := \min_{i \in [m]}\{\theta_i, 1 - \theta_i\}$.

The above proposition tells us that we can guarantee invariance for an arbitrary thin "tube" around the fixed components of the Boolean trap space if we choose the Hill coefficients sufficiently large. We will talk more in detail about this result in Section 2.4. Indeed it is even possible to give approximations on the size of $\vec{k_0}$ in Proposition 2.1 based on the Hill coefficients and the sparsity of the interaction graph of the BN.

Proposition 2.2. Let us denote with $d := \max_{x \in \{0,1\}^n, i \in [n]} \text{indeg}_{IG_f(x)}(i)$ the maximal indegree of $IG_f(x)$ over all $x \in \{0,1\}^n$ of a Boolean function $f \in \mathbb{B}(n,n)$. Assume $(p, *) \subseteq \{0,1\}^n$, $p \in \{0,1\}^m$, $m \leq n$, is a trap space of f. For

$$k_i \geq \begin{cases} \frac{\ln \epsilon - \ln(d-\epsilon)}{\ln \epsilon - \ln \theta_i}, & \text{if } p_i = 0 \\ \frac{\ln\left(\frac{d-\epsilon}{\epsilon}\right)}{\ln\left(\frac{1}{\theta_i} - \frac{1}{\theta_i}\epsilon\right)}, & \text{if } p_i = 1 \end{cases}$$

with $i \in [n]$, the set $K\big(p, \epsilon\big)$ is an invariant set of the ODE system (2.1.4) provided $\epsilon \in (0, \theta_{\text{min}})$.

We refer to Section 2.4.3 for details. Before we do that we show some applications of the above results.

2.3. Application: a small example

We want to illustrate how the theoretical results on trap spaces can be used to ease the analysis of ODE systems.

For this purpose, we consider an example from Klarner et al. [2015], namely a Boolean dynamical system with update function

$$f : (x_1, x_2, x_3, x_4) \mapsto (x_1 \vee x_2, x_1 \wedge x_4, \neg x_1 \wedge x_4, \neg x_3). \tag{2.3.1}$$

The state transition graph $G_{\text{sync}}(f)$ is depicted in Figure 2.3.1. As it can be seen in the figure, $(1, 1, 0, 1)$, $(1, *, 0, 1)$, $(1, *, 0, *)$, $(1, *, *, *)$ and $(0, 0, *, *)$ are trap spaces of the Boolean dynamical system given by f. Hence, the state of the first component plays a decisive role in the behavior of the system. Once the first component is activated, it will never be deactivated afterwards. According to the results presented in this chapter, this behavior should be similar in the corresponding ODE system as long as we choose sufficiently large Hill coefficients $\vec{k}$.

Example 2.5. Consider the trap space $(0, 0, *, *)$, where the maximal indegree of the local interaction graph is 2. Let $\vec{\theta} = (0.5, 0.5, 0.5, 0.5)^T$, $D = \operatorname{diag}(1, 1, 1, 1)$, and $\epsilon = 0.4$. We would like to find values for $\vec{k}$ such that $K\big(p, \epsilon\big)$ is an invariant set. In Figure 2.3.2 the conditions from Proposition 2.2 are depicted for the parameters $\theta = 0.5$ and maximal indegree $d = 2$. Since all fixed components in our trap space are zero, we only need to consider the curve corresponding to the first condition in Proposition 2.2. We can learn from this that we need to choose $k_i \geq 6.213$ to guarantee invariance for $\epsilon = 0.4$. More precisely, the first condition in Proposition 2.2 becomes:

$$k_i \geq \frac{\ln \epsilon - \ln(d - \epsilon)}{\ln \epsilon - \ln \theta_i} \geq \frac{\ln 0.4 - \ln(2 - 0.4)}{\ln 0.4 - \ln 0.5} \approx 6.213.$$

Indeed, if we choose an initial state $x_0 = (0.4, 0.4, 0.1, 0.1)^T$, and Hill coefficients $k_i \geq 6.213$, then the first two components tend to zero. Exactly this behavior can be observed in Figure 2.3.3a. Let us see what happens if we change the value of the first component slightly, such that the initial state is no longer in an invariant set corresponding to the trap space $(0, 0, *, *)$ but in one associated to $p = (1, *)$. For this purpose we could again consider Figure 2.3.2. Now we need to look at the curve corresponding to the second condition in Proposition 2.2, since all fixed components in the trap space p are 1. We can see that if we choose for example $\epsilon = 0.35$ the Hill exponent $k = 6.213$ is sufficiently large to guarantee that $K\big(p, \epsilon\big)$ is an invariant set of the corresponding ODE-system. More precisely, due to the inequality

$$6.213 \geq \frac{\ln\left(\frac{d-\epsilon}{\epsilon}\right)}{\ln\left(\frac{1}{\theta_i} - \frac{1}{\theta_i}\epsilon\right)} = \frac{\ln\left(\frac{2-0.35}{0.35}\right)}{\ln\left(\frac{1}{0.5} - \frac{1}{0.5} \cdot 0.35\right)},$$

we know that the trajectory of the solution is guaranteed to stay in $K\big(p, 0.35\big)$. Indeed, if we choose for example the initial value $x_0 = (0.65, 0.4, 0.1, 0.1)^T$, we observe in Figure 2.3.3b that the trajectory of the solution remains in

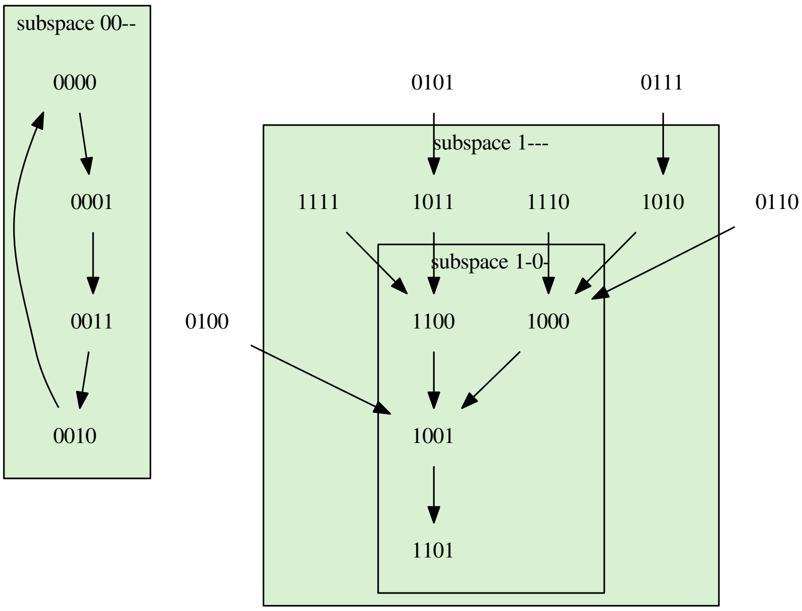


FIGURE 2.3.1. The graph $G_{\text{sync}}(f)$ with f defined in (2.3.1) (Example from Klarner et al. [2015]) containing the trap spaces $00{*}{*}$, $1{*}{*}{*}$ and $1{*}0{*}$ (which also contains $1{*}01$ and 1101 as trap spaces).

the set $K(p, \epsilon)$. Finally, let us consider a case, where we need to use both conditions from Proposition 2.2. I.e. for $p = (1, *, 0, *)$ to guarantee that $K(p, \epsilon)$ is an invariant set, we need to consider the maximum of both curves in Figure 2.3.2. Again we observe that for $\epsilon = 0.35$ the Hill exponents can remain at 6.213 to guarantee invariance.

For more numerical experiments we refer to the supplementary of Schwieger et al. [2018], where some experiments on a T-cell activation model were conducted. More numerical experiments concerning the conservation of trap spaces using normalized Hill cubes can be found in Zañudo and Albert [2015], too.

2.4. Correspondence of trap spaces in detail

We will give now proofs of Proposition 2.1 and Proposition 2.2. To do so we first consider the construction with normalized Hill functions. This is more easy since the continuous counterpart $\overline{f}$ of the Boolean function f agrees in this case with f on the vertices of the hypercube $\{0, 1\}^n$.

2.4.1. Correspondence of trap spaces for normalized Hill functions. We will show now that if we use normalized Hill functions, the trap spaces remain the same during the conversion from the Boolean model (2.1.1)

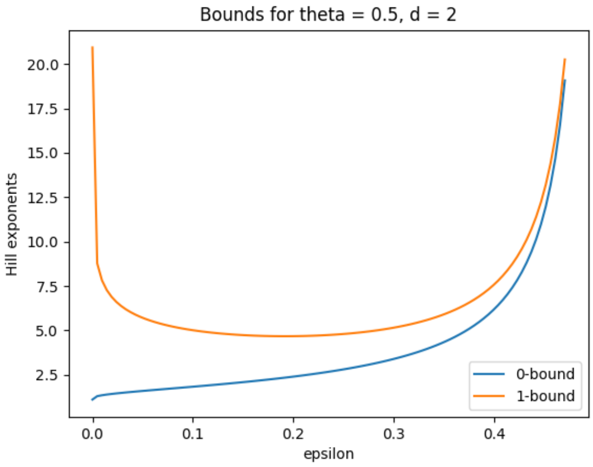


FIGURE 2.3.2. Plot of the conditions from Proposition 2.2. The curve denoted by 0-bound represents the condition in the first case in the proposition, while the second curve represents the condition in the second case. For an invariant set $K(p,\epsilon)$, corresponding to a trap space $(p,*)$, we can read off the minimal Hill exponent necessary for invariance from these curves.

to the ODE system

$$\begin{aligned} \dot{\overline{x}} &= D \cdot (\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}(\overline{x}) - \overline{x}), \\ \overline{x}(0) &= \overline{x}_0. \end{aligned} \tag{2.4.1}$$

The map $\frac{H^{\vec{k},\vec{\theta}}}{H^{\vec{k},\vec{\theta}}(\vec{1})}$ is a bijection from $[0,1]^n$ to $[0,1]^n$ for $\vec{k} \in \mathbb{R}^n_{>0}$ and $\vec{\theta} \in (0,1)^n$ and its restriction to $\{0,1\}^n$ is the identity map. It is also easy to show that $\overline{I}(f)$ is inheriting the trap spaces of f. We obtain:

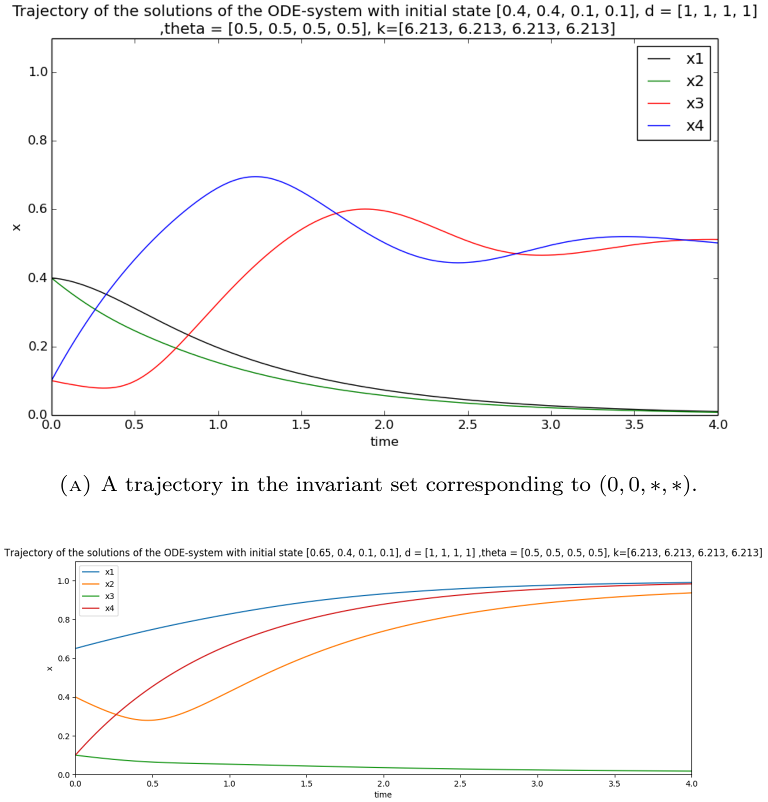
Proposition 2.3 ([Schwieger et al., 2018, Proposition 11])**.** Let $(p,*)$ be any trap space of $f \in \mathbb{B}(n,n)$, then $(\overline{p},*)$ with $\overline{p} = p \in \{0,1\}^m$, $m \in [n]$ is a trap space of $\overline{I}(f)$.

The idea of the proof of Proposition 2.3 is to show that if $p = 0$ holds then the corresponding components of the polynomial $\overline{I}(f_i)$, $i \in [m]$ are divisible by the monomials $\overline{x}_i$. The general case can then easily be derived of this special case by variable transformation. For the details we refer to [Schwieger et al., 2018, Proposition 11].

This proposition shows that we can assign to each trap space of f a trap space of $\overline{I}(f)$. This result then transfers to $\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}$.

Corollary 2.1. If $(p,*)$, $p \in \{0,1\}^m$ is a trap space of $f \in \mathbb{B}(n,n)$, then for any $\vec{k} \in \mathbb{R}^n_{>0}$ and $\vec{\theta} \in (0,1)^n$, $(p,*)$ is a trap space of the flow map defined by a solution of the ODE system of the form (2.4.1).

PROOF. Let $x : \mathbb{R}_{\geq 0} \to [0,1]^n$ be any solution of the above ODE system. Assume $x(t) \in (p,*)$, $t \in \mathbb{R}_{\geq 0}$. Since $(p,*)$ is a trap space of f, according to Proposition 2.3 $(p,*)$ is a trap space of $\overline{I}(f)$. The function $\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}$



(A) A trajectory in the invariant set corresponding to $(0, 0, *, *)$.

(B) A trajectory in the invariant set corresponding to $(1, *, *, *)$.

FIGURE 2.3.3. Illustration of Example 2.5.

is defined as the concatenation of normalized Hill cubes and $\overline{I}(f)$. Consequently, $(p, *)$ is a trap space of $\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}$. Therefore, for any $i \in [m]$ and $x(t) \in (p, *)$ the equality $\dot{x}_i(t) = d_i \cdot \left[(\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}})_i(x(t)) - p_i\right] = 0$ holds, and $(p, *)$ is an invariant set of any flow defined by the above ODE system. □

The above results say us nothing about the stability of the trap spaces in the ODE systems. If we want to know whether we will observe these trap spaces it is crucial to know what happens if the fixed components of these trap spaces are perturbed slightly. Therefore, we consider now an association to invariant sets that is possible even if we do not use normalized Hill functions.

2.4.2. Associating invariant sets to trap spaces. We want to show that $K(\overline{p}, \epsilon)$ is an invariant set of the ODE system (2.1.4) constructed with either $\overline{f}^{\vec{k},\vec{\theta}}$ or $\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}$ provided the Hill coefficients $\vec{k}$ are large enough. The proof is carried out in two steps. First, we show that $K(\overline{p}, \epsilon)$ is an invariant set of the time-discrete dynamical systems

$$\begin{aligned} x^{t+1} &= \overline{f}^{\vec{k},\vec{\theta}}(x^t), \\ x^{t+1} &= \overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}(x^t). \end{aligned} \tag{2.4.2}$$

Afterwards, we show the invariance for the corresponding ODE systems.

2.4.2.1. *Invariance in the time-discrete dynamical system.* Now we show that if $(\overline{p}, *)$ is a trap space of $\overline{f}^{\vec{k},\vec{\theta}}$ or $\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}$, then for sufficiently large Hill coefficients $\vec{k}$ the set $K(\overline{p}, \epsilon)$ is an invariant set of $\overline{f}^{\vec{k},\vec{\theta}}$ or $\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}$ as well.

Proposition 2.4. Let $f \in \mathbb{B}(n, n)$, L be the Lipschitz constant of $\overline{I}(f)$ with respect to $\|\cdot\|_\infty$ and $(\overline{p}, *)$ be a trap space of $\overline{I}(f)$. Then for any $0 < \gamma, \epsilon < \min_{i\in[m]}\{\theta_i, 1-\theta_i\}$ and sufficiently large $\vec{k}_0$, the function $\overline{f}^{\vec{k},\vec{\theta}}$ or, respectively, $\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}$ satisfies

$$\begin{aligned} \forall \vec{k} \geq \vec{k}_0 &: \overline{f}^{\vec{k},\vec{\theta}}(K(\overline{p}, \epsilon)) \subseteq K(\overline{p}, \gamma), \\ \forall \vec{k} \geq \vec{k}_0 &: \overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}(K(\overline{p}, \epsilon)) \subseteq K(\overline{p}, \gamma). \end{aligned} \tag{2.4.3}$$

The relation $\vec{k} \geq \vec{k}_0$ is meant component-wise here. This means, on the one hand, if $\vec{k}$ grows, there is an increasingly thin tube $K(\overline{p}, \gamma)$ around the trap spaces which is not left by the trajectories of the dynamical systems (2.4.2). On the other hand, there is an increasingly wide tube $K(\overline{p}, \epsilon)$, whose boundaries approach the thresholds of the Hill cubes, and every trajectory starting in $K(\overline{p}, \epsilon)$ is drawn into $K(\overline{p}, \gamma)$.

For the proof of Proposition 2.4 we exploit that $\overline{f}^{\vec{k},\vec{\theta}}$ as well as $\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}$ is a concatenation of the Hill functions and a multivariate polynomial interpolation. First, we prove that $K(\overline{p}, \epsilon)$ can be arbitrarily contracted by the Hill functions provided we choose the parameters $\vec{k}$ large enough and $K(\overline{p}, \epsilon)$ does not overlap with any of the thresholds of the Hill functions. Afterwards, we continue showing that the multivariate interpolation $\overline{I}(f)$ stretches the set $K(\overline{p}, \epsilon)$ maximally with a constant that is bounded by the Lipschitz constant of $\overline{I}(f)$.

Having the Hill functions in mind, we can show that for sufficiently large Hill coefficients the set $K(\overline{p}, \epsilon)$ can be arbitrarily contracted:

Lemma 2.1 ([Schwieger et al., 2018, Corollary 16])**.** For $m \leq n$ and $\vec{\theta} \in (0, 1)^n$, the following statement holds:

$$\forall \overline{p} \in \{0,1\}^m, \epsilon < \theta_{\min}, \gamma \in (0, 1] \exists \vec{k}_0 \in \mathbb{R}^n_{>0} \forall \vec{k} \geq \vec{k}_0 : H^{\vec{k},\vec{\theta}}\big(K(\overline{p}, \epsilon)\big) \subseteq K(\overline{p}, \gamma)$$

with $\theta_{\min} := \min_{i\in[m]}\{\theta_i, 1-\theta_i\}$. This remains true for normalized Hill cubes $\frac{H^{\vec{k},\vec{\theta}}}{H^{\vec{k},\vec{\theta}}(\vec{1})}$.

The proof of Lemma 2.1 is based on the observation that the Hill functions $h_{k,\theta}$ converge for increasing Hill coefficient k on the intervals $[0, a] \subset [0, \theta)$ uniformly towards zero and on the intervals $[b, 1] \subset (\theta, 1]$ towards one. Therefore, the Hill coefficients can always be chosen large enough such that

$K(\overline{p}, \epsilon)$ is contracted to $K(\overline{p}, \gamma)$ by the mapping $H^{\vec{k},\vec{\theta}}$. For details we refer to Schwieger et al. [2018].

We proceed with the multivariate interpolation of f.

Lemma 2.2 ([Schwieger et al., 2018, Lemma 17])**.** Let $\overline{I} : [0,1]^n \to [0,1]^n$ be a Lipschitz continuous function with respect to $\|\cdot\|_\infty$ with Lipschitz constant L. Furthermore, let $(\overline{p}, *)$ be a trap space of $\overline{I}$. Then

$$\forall \epsilon \in (0,1) : \overline{I}(K(\overline{p}, \epsilon)) \subseteq K(\overline{p}, L\epsilon)$$

holds.

Combining the two results Lemma 2.1 and Lemma 2.2 we can now prove Proposition 2.4 from the beginning of this section.

Proof. (Proof of Proposition 2.4) According to Lemma 2.1, if we choose $\gamma' \leq \frac{\gamma}{L}$ such that $\gamma' \in (0, \epsilon]$, we have for sufficiently large $\vec{k}$:

$$\begin{aligned} \overline{f}^{\vec{k},\vec{\theta}}(K(\overline{p}, \epsilon)) &= \overline{I}(f) \circ H^{\vec{k},\vec{\theta}}(K(\overline{p}, \epsilon)) \\ &\subseteq \overline{I}(f)(K(\overline{p}, \gamma')) \end{aligned}$$

According to Lemma 2.2, we have

$$\overline{I}(f)(K(\overline{p}, \gamma')) \subseteq K(\overline{p}, L\gamma') \subseteq K(\overline{p}, \gamma).$$

The statement $\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}(K(\overline{p}, \epsilon)) \subseteq K(\overline{p}, \gamma)$ follows analogously. □

2.4.2.2. *Invariance in the ODE system.* We show in this section the invariance of the set $K(\overline{p}, \epsilon)$ for a trap space $(\overline{p}, *)$ of the ODE system (2.1.4). The proof is carried out by exploiting the results on the time-discrete but state-continuous dynamical system of the previous section and combining them with a result by Nagumo on invariant sets Nagumo [1942].

A subset K of a finite dimensional vector space X is invariant (see Definition 1.20) under a dynamical system induced by the ODE

$$\begin{aligned} \dot{x} &= F(x), \\ x(0) &= x_0, \end{aligned} \tag{2.4.4}$$

if for any initial state $x_0 \in K$ all solutions to the differential equation (2.4.4) remain in K.

Definition 2.6 ([Aubin, 1991, Horváth et al., 2014, p. 25])**.** Let X be a normed space, $\emptyset \neq K \subseteq X$, $x \in X$. The *tangent cone* (Figure 2.4.1) to K at x is the set

$$T_K(x) := \{v \in X \,\big|\, \lim \inf_{h \to 0+} \frac{\inf_{z \in K} \|(x + hv) - z\|}{h} = 0\}.$$

We use the following Theorem from Nagumo [1942]:

Theorem 2.2 (Horváth et al. [2014], Nagumo [1942])**.** Let $K \subset \mathbb{R}^n$ be closed and convex, $F : \mathbb{R}^n \to \mathbb{R}^n$ continuous and assume (2.4.4) admits a globally unique solution for every $x_0 \in K$. Then K is an invariant set of this system if and only if

$$\forall x \in \partial K : F(x) \in T_K(x). \tag{2.4.5}$$

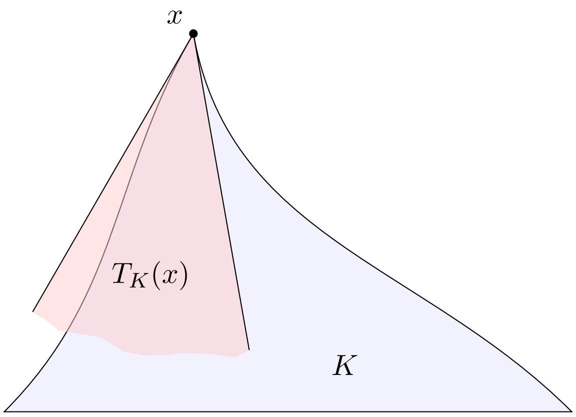


FIGURE 2.4.1. Illustration of the tangent cone $T_K(x)$ of K.

Now we are ready to prove Proposition 2.1 – the correspondence of trap spaces and invariant sets. That is:

Proposition 2.5. Let $(\overline{p}, *)$, $\overline{p} \in \{0,1\}^m$ be a trap space of $f \in \mathbb{B}(n,n)$. Then

$$\forall \epsilon \in (0, \theta_{\min}) \exists \vec{k_0} \in \mathbb{N}^n \forall \vec{k} \geq \vec{k_0} \forall x_0 \in K(\overline{p}, \epsilon) : \forall t \in [0, \infty) : x(t) \in K(\overline{p}, \epsilon),$$

where $x(t)$ is the solution of (2.1.4) and with $\theta_{\min} := \min_{i \in [m]}\{\theta_i, 1-\theta_i\}$.

PROOF. We want to show that the ODE system (2.1.4) with $K(\overline{p}, \epsilon)$ satisfies the conditions of Theorem 2.2. Here, our function F in Theorem 2.2 has the form:

$$F^{\vec{k},\vec{\theta}}(x) := D \cdot (\overline{f}^{\vec{k},\vec{\theta}}(x) - x)$$

We need to show that (2.4.5) with $K := K(\overline{p}, \epsilon)$ is satisfied. Let $\xi \in \partial K(\overline{p}, \epsilon)$. (2.4.5) holds especially true if $\xi + hF(\xi) \in K$ is satisfied for small enough $h > 0$. Therefore, it would suffice to show

$$\xi + h\big[D \cdot (\overline{f}^{\vec{k},\vec{\theta}}(\xi) - \xi)\big] = (1 - hD)\xi + hD \cdot \overline{f}^{\vec{k},\vec{\theta}}(\xi) \in K(\overline{p}, \epsilon).$$

Indeed, this is true for sufficiently small $h > 0$ (to guarantee $(1 - hd_i) > 0$ for $i \in [m]$) due to the following inequality for $i \in [m]$ and sufficiently large $\vec{k}$:

$$\begin{aligned} \|(1-hd_i)\xi_i + hd_i\overline{f}_i^{\vec{k},\vec{\theta}}(\xi) - \overline{p}_i\| &= \|(1-hd_i)(\xi_i - \overline{p}_i) + hd_i(\overline{f}_i^{\vec{k},\vec{\theta}}(\xi) - \overline{p}_i)\| \\ &\leq (1-hd_i) \cdot \underbrace{\|\xi_i - \overline{p}_i\|}_{\leq \epsilon} + hd_i \cdot \underbrace{\|\overline{f}_i^{\vec{k},\vec{\theta}}(\xi) - \overline{p}_i\|}_{\leq \epsilon} \\ &\leq \epsilon \qquad (2.4.6) \\ \Rightarrow \xi + hD \cdot (\overline{f}^{\vec{k},\vec{\theta}}(\xi) - \xi) &\in K(\overline{p}, \epsilon), \end{aligned}$$

where the inequality $\|\overline{f}_i^{\vec{k},\vec{\theta}}(\xi) - \overline{p}_i\| \leq \epsilon$ follows from Proposition 2.4. The uniqueness of the solution of the ODE system (2.1.4) follows from the Lipschitz continuity of $\overline{f}^{\vec{k},\vec{\theta}}$ (see, e.g., [Sastry, 2013, p. 88]). □

Again, the same argumentation is valid if we replace $\overline{f}^{\vec{k},\vec{\theta}}$ by $\overline{f}^{\vec{k},\vec{\theta}}_{\text{normalized}}$.

2.4.3. Computing explicit values for the Hill coefficients to guarantee invariance. Finally, we are interested in finding some explicit boundaries for the values $\vec{k}$ and to find out how they are related to the network structure. Namely, assume we would like to find Hill coefficients $\vec{k}$ such that $K(p,\epsilon)$ is mapped to $K(p,\gamma)$ by $\overline{f}^{\vec{k},\vec{\theta}}$ for given $\epsilon,\gamma > 0$ in such a way that $K(p,\epsilon)$ is an invariant set of the corresponding ODE system.

We can approximate the Lipschitz constants of $\overline{I}(f)$ explicitly. First, notice that if we want to find the Lipschitz constant L of $\overline{I}(f) : [0,1]^n \to [0,1]^n$, i.e.

$$\forall x, y \in [0,1]^n : \|\overline{I}(f)(x) - \overline{I}(f)(y)\|_\infty \leq L \cdot \|x-y\|_\infty, \tag{2.4.7}$$

we can do this by finding the Lipschitz constants of the components of $\overline{I}(f) = (\overline{I}(f_1), \dots, \overline{I}(f_n))$ and then taking the maximal Lipschitz constant for $\overline{I}(f)$. For the Lipschitz constants of the components $\overline{I}(f_i)$ we find an upper bound with the following theorem:

Theorem 2.3 ([Paulavičius and Žilinskas, 2006, Theorem 1])**.** Assume $\overline{f} : [0,1]^n \to [0,1]$ is a Lipschitz continuous function. Then

$$|\overline{f}(x) - \overline{f}(y)| \leq L_p \cdot \|x-y\|_q \tag{2.4.8}$$

and

$$L_p = \sup\{\|\nabla \overline{f}(x)\|_p : x \in [0,1]^n\}$$

with $\frac{1}{p} + \frac{1}{q} = 1$, $1 \leq p, q \leq \infty$.

Applying the theorem to the case $q = \infty$ we need to find $\sup_{x\in[0,1]^n} \|\nabla \overline{I}(f_i)\|_1$ for all $i \in [n]$.

After having estimated the Lipschitz constant L of $\overline{I}(f)$, we can proceed as follows to obtain values for $\vec{k}$ that guarantee that $K(p,\epsilon)$ is mapped to $K(p,\gamma)$ for a trap space $(p,*)$ of f. For fixed $\vec{\theta}$ we need to find k_i such that

$$\begin{aligned} h_{k_i,\theta_i}(\epsilon) &\leq \frac{\epsilon}{L} \text{ if } p_i = 0, \\ h_{k_i,\theta_i}(1-\epsilon) &\geq 1 - \frac{\epsilon}{L} \text{ if } p_i = 1 \end{aligned} \tag{2.4.9}$$

holds to guarantee $H^{\vec{k},\vec{\theta}}\big(K(\overline{p},\epsilon)\big) \subseteq K(\overline{p},\frac{\epsilon}{L})$. Then Lemma 2.2 implies

$$\overline{f}^{\vec{k},\vec{\theta}}(K(\overline{p},\epsilon)) \subseteq \overline{I}(f)\big(K(\overline{p},\frac{\epsilon}{L})\big) \subseteq K(\overline{p},\epsilon).$$

While (2.4.9) is independent of the structure of the bioregulatory system, the size of the Lipschitz constant of $\overline{I}(f)$ is not. Therefore, it is of interest to find out in what respect it depends on the structure of the discrete interaction graph.

In Schwieger et al. [2018] the following result was obtained.

Proposition 2.6 ([Schwieger et al., 2018, Corollary 31])**.** Let $IG_f(x)$ be the local interaction graph of $f : \{0,1\}^n \to \{0,1\}^n$. Then $\max_{x\in\{0,1\}^n} \|\nabla f_i(x)\|_1$ is the maximal indegree of $i \in V = [n]$ over all local interaction graphs, i.e.,

$$\max_{x\in\{0,1\}^n} \operatorname{indeg}_{IG_f(x)}(i) = \max_{x\in\{0,1\}^n} \|\nabla f_i(x)\|_1.$$

Furthermore, the Lipschitz constant L of $\overline{I}(f)$ can be approximated by the maximum degree over all local interaction graphs of f,[14] i.e.,

$$\max_{x\in\{0,1\}^n, i\in[n]} \operatorname{indeg}_{IG_f(x)}(i) \geq L, \tag{2.4.10}$$

where L is the Lipschitz constant of $\overline{I}(f)$.

Consider an arbitrary Boolean function $f : \{0,1\}^n \to \{0,1\}^n$ with interaction graph $IG_f(x)$, $x \in \{0,1\}^n$ and a trap space $(p,*) \subseteq \{0,1\}^n$, $p \in \{0,1\}^m$, $m \leq n$. We give now the condition already mentioned (Proposition 2.2) on the parameters $\vec{\theta}$ and $\vec{k}$ of the ODE system (2.1.4) which guarantees that $K(p,\epsilon)$ is an invariant set of (2.1.4).

Proposition 2.7. Let us denote with $d := \max_{x\in\{0,1\}^n, i\in[n]} \operatorname{indeg}_{IG_f(x)}(i)$ the maximal indegree of $IG_f(x)$ over all $x \in \{0,1\}^n$ of a Boolean function $f : \{0,1\}^n \to \{0,1\}^m$. Assume $(p,*) \subseteq \{0,1\}^n$, $p \in \{0,1\}^m$, $m \leq n$, is a trap space of f. For

$$k_i \geq \begin{cases} \frac{\ln\epsilon - \ln(d-\epsilon)}{\ln\epsilon - \ln\theta_i}, & \text{if } p_i = 0 \\ \frac{\ln\left(\frac{d-\epsilon}{\epsilon}\right)}{\ln\left(\frac{1}{\theta_i} - \frac{1}{\theta_i}\epsilon\right)}, & \text{if } p_i = 1 \end{cases}$$

with $i \in [n]$, the set $K\big(p,\epsilon\big)$ is an invariant set of the ODE system (2.1.4) provided $\epsilon \in (0, \theta_{\min})$.

Proof. Due to Proposition 2.6, the Lipschitz constant L of $\overline{I}$ with respect to $\|\cdot\|_\infty$ can be approximated by d. Therefore, it remains to prove

$$H^{\vec{k},\vec{\theta}}\big(K(p,\epsilon)\big) \subseteq K(p, \frac{\epsilon}{d}).$$

We need to check for each component of $H^{\vec{k},\vec{\theta}}$ the condition

$$\begin{cases} h_{k_i,\theta_i}(\epsilon) \leq \frac{\epsilon}{d}, & \text{if } p_i = 0 \\ h_{k_i,\theta_i}(1-\epsilon) \geq 1 - \frac{\epsilon}{d}, & \text{if } p_i = 1 \end{cases}.$$

For $i \in [m]$ we obtain for $p_i = 0$ the condition

$$\begin{aligned} h_{k_i,\theta_i}(\epsilon) &\leq \frac{\epsilon}{d} \\ \Leftrightarrow \frac{\epsilon^{k_i}}{\epsilon^{k_i} + \theta_i^{k_i}} &\leq \frac{\epsilon}{d} \\ \Leftrightarrow k_i \cdot \underbrace{\ln\frac{\epsilon}{\theta_i}}_{<0} &\leq \ln\frac{\epsilon}{d-\epsilon} \\ \Leftrightarrow k_i &\geq \frac{\ln\frac{\epsilon}{d-\epsilon}}{\ln\frac{\epsilon}{\theta_i}} = \frac{\ln\epsilon - \ln(d-\epsilon)}{\ln\epsilon - \ln\theta_i} \end{aligned}$$

[14] This implies that $\max_{x\in\{0,1\}^n} \|\nabla f_i(x)\|_1$ is bounded by the maximal indegree of the global interaction graph as well.

and for $p_i = 1$ the condition

$$\begin{aligned} h_{k_i,\theta_i}(1-\epsilon) &\geq 1 - \frac{\epsilon}{d} \\ \Leftrightarrow k_i &\geq \frac{\ln\left(\frac{d-\epsilon}{\epsilon}\right)}{\ln\left(\frac{1}{\theta_i} - \frac{1}{\theta_i}\epsilon\right)} \end{aligned}$$

□

We can obtain a similar result for the normalized ODE system.

2.5. Discussion

Firstly, we saw in this chapter that statements about the correspondence between steady states of discrete dynamical systems and corresponding continuous dynamical systems cannot be generalized directly to trap spaces (Example 2.4). However, in case of ODE systems created from Boolean functions by multivariate polynomial interpolation and Hill cubes, Boolean trap spaces can be linked to invariant sets (Proposition 2.1). Secondly, we saw that their size can be controlled by the Hill coefficients. We gave lower bounds for the Hill coefficients to guarantee invariance (Proposition 2.2).

Finally, we demonstrated experimentally in Section 2.3 that this can be exploited to shift a part of the analysis of the ODE-system into the analysis of the initial Boolean model. In Schwieger et al. [2018] this was also demonstrated in a larger example modeling T-cell activation.

The results here open the door for the following work flow:

(1) Start with a Boolean model.
(2) Identify trap spaces and use these trap spaces to generate reduced Boolean models.
(3) Convert the reduced Boolean models into ODE models and use quantitative data to fit the ODE models.
(4) Optional: Use the parameters obtained in this way to generate an ODE model of the complete regulatory network.

Switching between the different formalisms or combining them in hybrid models is of great benefit for many applications where the available data sets are highly inhomogeneous, containing data on different levels of resolution. The Boolean model can be considered as a map that gives us a coarse understanding of the regulatory network. Then if necessary, we can “zoom” into parts of this map by generating an ODE model. This allows for a better understanding of crucial parts of the regulatory networks. In this way the use of Boolean models can guide the research and successive modeling of a network in situations where modeling the complete regulatory network with ODEs is impossible.

In addition, logical models are much more accessible for non-experts, making it convenient to use them for testing hypotheses that can then be transferred to more detailed models. The results in this chapter represent an approach that goes beyond heuristics, since the conservation of certain mathematical structures – trap spaces – and properties across the two formalisms considered here can be proven.

However, it would be beneficial to extend the properties that are guaranteed to be transfered from the BN to the ODE model to properties relying on paths in the ASTG or STG. This would give us a justification to look at reachability properties of the ASTG. The results here are more or less independent of the exact nature of the update-rule and the state transition graph of the BN.

CHAPTER 3

Operations on Boolean networks

Contents

3.1. Set operations on ASTGs **46**
3.2. Quotient graphs of Boolean networks **49**
3.2.1. Projections 52
3.2.1.1. Compact representation of projected ASTGs 53
3.2.1.2. Some remarks on conditions for equality in Proposition 3.3 55
3.2.1.3. Relation between projections, subspaces and trap spaces 56
3.2.2. The quotient graph with respect to the update function 57
3.3. Inverting transitions **60**
3.4. Discussion **60**

In this chapter we introduce several operations on BNs. These operations will be used in the following chapters. In the most simple case in Section 3.1 we consider how simple graph operations such as intersection and union can be realized using the update function. We will make use of this in the context of experiment design in Chapter 5. In Section 3.2 we will consider several quotient graphs - such as quotient graphs introduced by projections or the update function of the (asynchronous) BN itself. With quotient graphs it is possible to summarize several states of a BN into a single node. They can be seen as a way to abstract the dynamics of a BN. Therefore, it makes sense to use them in the context of model pools (see Chapter 4). We consider the operations here mainly from a theoretical point of view as a tool in our proofs. Therefore, computational aspects are neglected. Nevertheless, if possible we describe methods to perform the operations directly in terms of the update function of the asynchronous BN. This should be, at least in theory, computationally beneficial, since the methods we suggest avoid the explicit construction of the ASTG.

3.1. Set operations on ASTGs

In this section we discuss various simple set operations on state transition graphs – more precisely ASTGs. The need for performing such operations can have different motivations. In general, the state transition graphs of our Boolean models describe potential behaviors of the modeled GRN. Combining different sources of knowledge or comparing these models (see also Section 5.1) can in some instances be translated into comparing the state transition graphs of different BNs. Such comparisons in general rely on set operations of the state transition graphs:

Definition 3.1 (see e.g. [Ruohonen, 2013, p. 11])**.** We are given two graphs $G^1 = (V^1, E^1)$ and $G^2 = (V^1, E^2)$.[1] Then we define:

$$\begin{aligned} G^1 \cup G^2 &:= (V^1, E^1 \cup E^2), \\ G^1 \cap G^2 &:= (V^1, E^1 \cap E^2), \\ G^1 \backslash G^2 &:= (V^1, E^1 \backslash E^2), \\ G^1 \triangle G^2 &:= G^1 \backslash G^2 \cup G^2 \backslash G^1. \end{aligned}$$

From a practical point of view these operations can be performed by enumerating and comparing all edges of both graphs. In the case of Boolean networks the transitions of the state transition graph are implicitly given by a Boolean function. To avoid explicit enumeration of transitions it is preferable to perform the above set operations directly on the Boolean function inducing the state transition graph. We show here how this can be done for asynchronous BNs.

For this imagine the case where $G^1 = G_{\text{async}}(f)$ and $G^2 = G_{\text{async}}(g)$ are induced by two Boolean functions $f, g \in \mathbb{B}(n, n)$. To perform the set operations from Definition 3.1 directly on the Boolean representatives of these graphs we introduce a function $\text{cond}^f \in \mathbb{B}(n, n)$ associated to f, which tells us if there is a transition between two states. The function cond^f will be constructed in such a way that there is a transition $(x, x^{\{i\}}) \in E_{\text{async}}(f)$, $i \in [n]$ if and only if $\text{cond}^f_i(x) = 1$.

Definition 3.2. For a Boolean function $f : \{0,1\}^n \to \{0,1\}^n$ we define $\text{cond}^f : \{0,1\}^n \to \{0,1\}^n, x \mapsto \text{cond}^f(x)$ by $\text{cond}^f(x) := \text{id} \oplus f(x)$, where $\text{id} : \{0,1\}^n \to \{0,1\}^n$ denotes the identity function.

For a state $x \in \{0,1\}^n$ the function cond^f evaluates to one in a component $i \in [n]$ if and only if the value of $f_i(x)$ and the value of the state x differ. The function cond^f indicates if a change in a component is possible. More precisely, we have the following lemma:

Lemma 3.1. There is a transition $(x, x^{\{i\}}) \in E_{\text{async}}(f)$, $i \in [n]$ if and only if $\text{cond}^f_i(x) = 1$.

PROOF. Consider

$$\begin{aligned} (x, x^{\{i\}}) \in E_{\text{async}}(f) \Leftrightarrow & i \in \text{diff}(x, f(x)) \\ \Leftrightarrow & x_i \oplus f_i(x). \end{aligned}$$

□

It is also easy to see that for $f \in \mathbb{B}(n, n)$ it holds $f = \text{cond}^{\text{cond}^f}$. Furthermore, it is clear that the function $\text{cond}^{\cdot} : \mathbb{B}(n, n) \to \mathbb{B}(n, n)$ is a bijection. Similar to Lemma 3.1 we can represent a graph $G = (\{0,1\}^n, E)$ constructed from a Boolean function $g : \{0,1\}^n \to \{0,1\}^n$ using the component functions of g as indicator functions for the transitions in E, as an ASTG of a Boolean function:

[1] We assume that the set of nodes in both graphs is the same. This simplifying assumption is made in view of of their applicability to ASTGs in the next section.

$$\begin{tikzcd}
\operatorname{cond}^f \in \mathbb{B}(n,n) \arrow[dr, dashed, "\operatorname{cond}^f \text{ as indicator function}"] \arrow[dd, dashed, bend left, "\operatorname{cond}^{\operatorname{cond}^f}"] & \\
& G_{\text{async}}(f) \\
f(=\operatorname{cond}^{\operatorname{cond}^f}) \in \mathbb{B}(n,n) \arrow[uu, dashed, bend left, "\operatorname{cond}^f"] \arrow[ur, dashed, "f \text{ as update function}"] &
\end{tikzcd}$$

FIGURE 3.1.1. The function $\operatorname{cond}^{\cdot} : \mathbb{B}(n,n) \to \mathbb{B}(n,n), f \mapsto \operatorname{cond}^f$ is a bijective map from $\mathbb{B}(n,n)$ to $\mathbb{B}(n,n)$. For a function $f \in \mathbb{B}(n,n)$ the graph $G_{\text{async}}(f)$ can either be induced by the update function f or by the condition function cond^f.

Lemma 3.2. For a graph $G = (V, E)$ defined by $V = \{0,1\}^n$ and

$$(v, v^{\{i\}}) \in E :\Leftrightarrow g_i(v),$$

for $i \in [n]$ and $g \in \mathbb{B}(n,n)$ it holds $G_{\text{async}}(\operatorname{cond}^g) = G$.

PROOF. This is a consequence of Lemma 3.1. Choose $f := \operatorname{cond}^g$. Then due to the identity $g = \operatorname{cond}^{\operatorname{cond}^g} = \operatorname{cond}^f$ and Lemma 3.1 the transitions of $G_{\text{async}}(f)$ are indicated by g, which shows equality. □

So in conclusion, the function cond^f is just another way to represent the ASTG $G_{\text{async}}(f)$. The relation of f and cond^f is illustrated in Figure 3.1.1. This observation will be frequently useful. Using this function it is easy to express common set operations of ASTGs:

Proposition 3.1. It holds

$$G_{\text{async}}(f) \cup G_{\text{async}}(g) = G_{\text{async}}(\operatorname{cond}^{\operatorname{cond}^f \vee \operatorname{cond}^g}), \tag{3.1.1}$$

$$G_{\text{async}}(f) \cap G_{\text{async}}(g) = G_{\text{async}}(\operatorname{cond}^{\operatorname{cond}^f \wedge \operatorname{cond}^g}), \tag{3.1.2}$$

$$G_{\text{async}}(f) \backslash G_{\text{async}}(g) = G_{\text{async}}(\operatorname{cond}^{\operatorname{cond}^f \wedge \neg\operatorname{cond}^g}), \tag{3.1.3}$$

$$G_{\text{async}}(f) \triangle G_{\text{async}}(g) = G_{\text{async}}(\operatorname{cond}^{f \oplus g}), \tag{3.1.4}$$

$$G_{\text{async}}(\neg\text{id}) \backslash G_{\text{async}}(f) = G_{\text{async}}(\neg f). \tag{3.1.5}$$

PROOF. Using Lemma 3.2 it holds for $i \in [n]$:

(3.1.1): $(s, s^{\{i\}}) \in E_{\text{async}}(f) \cup E_{\text{async}}(g) \Leftrightarrow \big[(s, s^{\{i\}}) \in E_{\text{async}}(f)$ or $(s, s^{\{i\}}) \in E_{\text{async}}(g)\big]$
$\Leftrightarrow \operatorname{cond}_i^f(s) \vee \operatorname{cond}_i^g(s)$

(3.1.2): $(s, s^{\{i\}}) \in E_{\text{async}}(f) \cap E_{\text{async}}(g) \Leftrightarrow \big[(s, s^{\{i\}}) \in E_{\text{async}}(f)$ and $(s, s^{\{i\}}) \in E_{\text{async}}(g)\big]$
$\Leftrightarrow \operatorname{cond}_i^f(s) \wedge \operatorname{cond}_i^g(s)$

(3.1.3): $(s, s^{\{i\}}) \in E_{\text{async}}(f) \backslash E_{\text{async}}(g) \Leftrightarrow \big[(s, s^{\{i\}}) \in E_{\text{async}}(f)$ and $(s, s^{\{i\}}) \notin E_{\text{async}}(g)\big]$
$\Leftrightarrow \operatorname{cond}_i^f(s) \wedge \neg\operatorname{cond}_i^g(s)$

(3.1.4):$(s, s^{\{i\}}) \in E_{\text{async}}(f) \triangle E_{\text{async}}(g) \Leftrightarrow \left[[(s, s^{\{i\}}) \in E_{\text{async}}(f)] \oplus [(s, s^{\{i\}}) \in E_{\text{async}}(g)]\right]$ $\Leftrightarrow \text{cond}_i^f(s) \oplus \text{cond}_i^g(s)$.
(3.1.5):$(s, s^{\{i\}}) \in E_{\text{async}}(\neg\text{id}) \setminus E_{\text{async}}(f) \Leftrightarrow (s, s^{\{i\}}) \in E_{\text{async}}(\text{id} \oplus [\text{cond}^{\neg\text{id}} \wedge \neg\text{cond}^f]) =$ $E_{\text{async}}(\text{id} \oplus \neg\text{cond}^f) = E_{\text{async}}(\underbrace{\text{id} \oplus \neg\text{id}}_{=1} \oplus f) = E_{\text{async}}(\neg f)$.

The above equalities follow then using Lemma 3.1. For (3.1.4) consider the equality

$$\begin{aligned} \text{cond}_i^f(s) \oplus \text{cond}_i^g(s) &= s_i \oplus \text{cond}_i^f(s) \oplus s_i \oplus \text{cond}_i^g(s) \\ &= f_i(s) \oplus g_i(s). \end{aligned}$$

□

Example 3.1. Let us consider the two Boolean functions:

$$\begin{aligned} f :& \{0,1\}^3 \to \{0,1\}^3, \\ & (x_1,, x_2, x_3) \mapsto (\neg x_3, x_1 \vee x_2 \vee x_3, x_2 \wedge \neg x_3), \end{aligned}$$

and

$$\begin{aligned} g :& \{0,1\}^3 \to \{0,1\}^3, \\ & (x_1,, x_2, x_3) \mapsto (f_1(x), f_2(x), x_2 \vee \neg x_3). \end{aligned}$$

Now let us look at $G := G_{\text{async}}(f) \triangle G_{\text{async}}(g)$. Using Proposition 3.1 we conclude that we can represent G as an ASTG with a Boolean function $h := \text{cond}^{f \oplus g}$, i.e. $G = G_{\text{async}}(h)$. Due to $\text{cond}_1^f = \text{cond}_1^g$, $\text{cond}_2^f = \text{cond}_2^g$ it holds

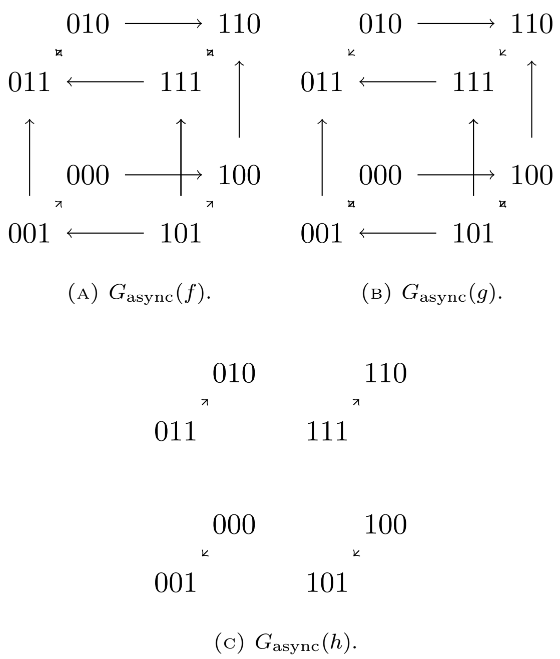
$$\begin{aligned} h_1 =& \text{id} \oplus 0 = \text{id}, \\ h_2 =& \text{id} \oplus 0 = \text{id}, \\ h_3(x) =& x_3 \oplus f_3(x) \oplus g_3(x) = \neg x_2. \end{aligned}$$

The ASTGs of f, g and h are depicted in Figure 3.1.2.

3.2. Quotient graphs of Boolean networks

Sometimes we are not interested in the specific single states of an ASTG, but rather in different groups of states sharing some property. Such properties could be for example that the states in one group belong to the same subspace or are reachable from each other. Quotient graphs are a common formalism for such abstractions of graphs. States which share a common property are referred to as equivalent states. Groups of states with the same properties are equivalence classes in this formalism. We then construct a new graph, the quotient graph, which has nodes representing equivalence classes and edges between these nodes, when the underlying equivalent classes are connected by an edge somewhere. In the following, we give a formal definition.

Consider an arbitrary directed graph $G = (V, E)$ and an equivalence relation $\sim$ on its nodes V. For a node $v \in V$ we denote with $[v]_\sim$ its equivalence class of v with respect to $\sim$, i.e. the set of nodes equivalent to v. If it is clear from the context which equivalence relation is meant we write $[v]$ instead of $[v]_\sim$. The relation $\sim$ can be used to construct a new graph from G, the so-called quotient graph:



(A) $G_{\text{async}}(f)$.

(B) $G_{\text{async}}(g)$.

(C) $G_{\text{async}}(h)$.

FIGURE 3.1.2. The graphs $G_{\text{async}}(f), G_{\text{async}}(g)$ and $G_{\text{async}}(h)$ in Example 3.1.

Definition 3.3 (see e.g. [Hahn and Tardif, 1997, Definition 2.8])**.** For any graph $G = (V, E)$ and an equivalence relation $\sim$ on the set of nodes V the *quotient graph* $G\big/\sim$ is defined by

$$G\big/\sim = (\tilde{V}, \tilde{E})$$

with $\tilde{V} := \big\{[v] | v \in V\big\}$ and $\tilde{E} := \big\{([v],[w]) | \exists \tilde{v} \in [v], \tilde{w} \in [w] : (\tilde{v}, \tilde{w}) \in E\big\}$.

Example 3.2. Consider the equivalence class induced by

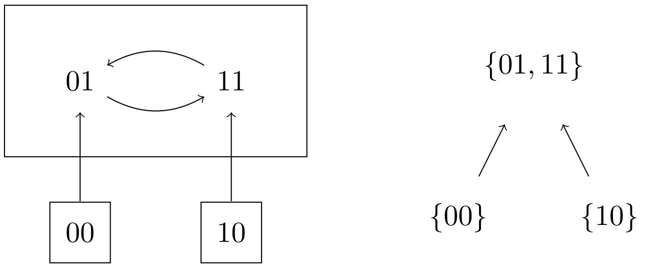
$$v \sim w :\Leftrightarrow (v \rightsquigarrow w \text{ and } w \rightsquigarrow v),$$

illustrated in Figure 3.2.1. I.e. two nodes are equivalent if and only if they are reachable from each other. The quotient graph is then the graph of strongly connected components.

We will refer to the nodes of a quotient graph derived from an ASTG also as states and to its edges also as transitions. Quotient graphs are interesting with respect to path connectedness. Any negative statement about the path-connectedness of two states in the quotient graph can be lifted up to the original graph.

Proposition 3.2. Let $\sim$ be any equivalence relation on the graph $G = (V, E)$. If there is no path from $[v]$ to $[w]$ in $G\big/\sim$ then there is no path from v to w in G.

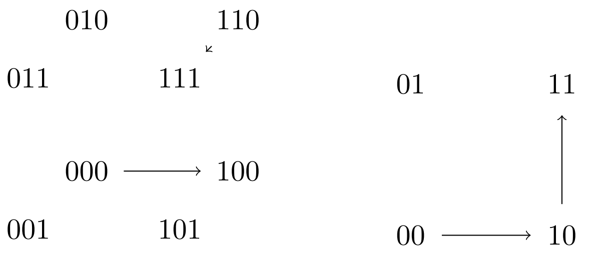
The other direction in Proposition 3.2 is not necessarily true. We demonstrate this with the following example:



(A) There are three SCCs: $\{01, 11\}$ and $\{00\}$ and $\{10\}$.

(B) The resulting SCC-graph.

FIGURE 3.2.1. Illustration of the SCC-graph in Example 3.2.



(A) $G_{\text{async}}(f)$: 000 and 111 are not path connected.

(B) $G_{\text{async}}(\text{proj}^{\{1,3\}}(f))$. $\text{proj}^{\{1,3\}}(000)$ and $\text{proj}^{\{1,3\}}(111)$ are path-connected.

FIGURE 3.2.2. Illustration of Example 3.3.

Example 3.3. Consider the ASTG depicted in Figure 3.2.2a. We see that the states 000 and 111 are not path-connected. We now consider the quotient graph induced by a projection onto the first and third component. Two states are equivalent if they are projected to the same state (see Section 3.2.1 for details.). Each equivalence class is represented by its image under the projection. In Figure 3.2.2b this quotient graph is depicted. We see that the states 00 and 11 are connected by a path. However, there was no path from 000 to 111 in the original graph.

Equivalence classes can be induced by functions of the form $\varphi : V \to \tilde{V}$. In the next section we will give an example using projections.

Definition 3.4. For a function $\varphi : V \to \tilde{V}$ we define

$$\begin{aligned} v &\sim_{\varphi} w \\ &:\Leftrightarrow \varphi(v) = \varphi(w). \end{aligned}$$

We write G/φ for the quotient graph $G/\sim_{\varphi}$.

We can identify the equivalence classes induced by $\sim_{\varphi}$ with their images under φ. Therefore, we can name the states in the quotient graphs by the corresponding image under φ, i.e. we write for a surjective function $\varphi : V \to \tilde{V}$

$$G/\sim_{\varphi} = (\tilde{V}, \tilde{E}).$$

The states $v \in \tilde{V}$ of the quotient graph $G/\sim_{\varphi}$ correspond to the equivalence classes $\varphi^{-1}(v)$. In the following sections we see two such examples. In the first example we consider projections. In this case we can refer to the set of states projected to the same components simply as their projected state.

3.2.1. Projections. We study now the case where the equivalence classes are induced by projections on some of the components of the Boolean network. Since in many biological applications only the values of some components matter or can be measured, it is useful to study these maps. In Section 5.1 for example we will study different formalism involving projections to design experiments to distinguish different models. In this case we project our model on the components we want to or can measure and investigate subsequently properties of the projected model that can be verified or falsified. As usual, due to the state explosion of Boolean models, our goal is to perform the projection of the model in such a way that no explicit enumeration of the state space is necessary. Instead we want to give an implicit description in terms of the Boolean function $f \in \mathbb{B}(n,n)$ inducing the Boolean network. Indeed, we see that this is partially possible.

Let $\emptyset \neq I \subseteq [n]$ be a set of components of a Boolean network. If not further specified we will use the letter I for a non-empty subset of $[n]$. Let us recall the definition of a projection (see Definition 1.26):

Definition. For any $\emptyset \neq I \subseteq [n]$ we defined the map $\mathrm{proj}^I : \{0,1\}^n \to \{0,1\}^{|I|}, x \mapsto (x_i)_{i \in I}$ and called it *projection*.

The map proj^I projects every state in $\{0,1\}^n$ onto the components in $\emptyset \neq I \subseteq [n]$. Now we introduce the projection of a state transition graph $G = (\{0,1\}^n, E)$ as the quotient graph using Definition 3.4.

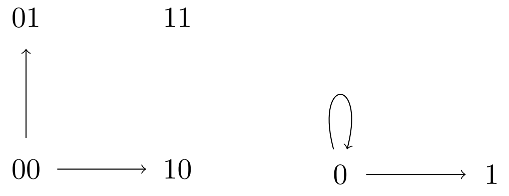
Definition 3.5. For a graph $G = (\{0,1\}^n, E)$ we call G/proj^I the projection of the graph G on I.

Definition 3.6. For an ASTG $G_{\mathrm{async}}(f)$ given by a Boolean function $f \in \mathbb{B}(n,n)$ and $\emptyset \neq I \subseteq [n]$ we call $G_{\mathrm{async}}(f)/\mathrm{proj}^I$ the projection of the ASTG on I.

There is a transition between any two states s, t in $G_{\mathrm{async}}(f)/\mathrm{proj}^I$ if and only if there is a transition between two states v, w in $G_{\mathrm{async}}(f)$ and v is projected to s and w is projected to t. We emphasize that this holds not necessarily true for paths. We mean with this that it could be that t is reachable from s in $G_{\mathrm{async}}(f)/\mathrm{proj}^I$ but there is no pair of states v, w in $G_{\mathrm{async}}(f)$ from the corresponding equivalence classes of s and t such that $v \rightsquigarrow w$ (see Example 3.3).

We illustrate the construction of the quotient graph $G_{\mathrm{async}}(f)/\mathrm{proj}^I$ with a small example:

Example 3.4. Take the ASTG $G_{\mathrm{async}}(f)$ induced by the function $f : (x_1, x_2) \mapsto (x_1 \oplus [\neg x_1 \wedge \neg x_2], x_2 \oplus [\neg x_1 \wedge \neg x_2])$. Its ASTG is depicted in Figure 3.2.3. We see that this graph has the transitions 00 → 10 and



(A) $G_{\text{async}}(f)$.

(B) Projection of $G_{\text{async}}(f)$ onto $\{1\}$.

FIGURE 3.2.3. Illustration of Example 3.4. The equivalence classes in the graph $G_{\text{async}}(f)/\text{proj}^{\{1\}}$ are identified with the images of $\text{proj}^{\{1\}}$.

$00 \to 01$. If we project on the first component – that means we consider the graph $G_{\text{async}}(f)/\text{proj}^{\{1\}}$ – we have the transitions $0 \to 1$ and $0 \to 0$. This means the state (0) has an outgoing transition and a loop. Consequently it cannot be an ASTG.

3.2.1.1. *Compact representation of projected ASTGs.* As emphasized in Section 3.1, in order to avoid enumeration of the states and transitions in $G_{\text{async}}(f)$ it would be desirable to have an implicit representation of the quotient graph $G_{\text{async}}(f)/\text{proj}^I$ in terms of the update function f. Therefore, we pose the question if we can derive from the representation f of the graph $G_{\text{async}}(f)$ some compact representation of $G_{\text{async}}(f)/\text{proj}^I$ as an ASTG.

However, we already saw in Example 3.4 that such a Boolean function cannot exist in general since the projection of an ASTG is not necessarily anymore an ASTG. However, if we delete loops from the quotient graph the projection of an ASTG is itself an ASTG and we can derive the function representing this graph without constructing the complete graph explicitly. For this purpose we will define a projection proj^I of a Boolean function in such a way that for a Boolean function $\varphi \in \mathbb{B}(n,1)$ the projected Boolean function $\text{proj}^I(\varphi) \in \mathbb{B}(|I|,1)$ will evaluate to one in a state $x \in \{0,1\}^n$ if and only there exists a state $\tilde{x} \in (\text{proj}^I)^{-1}(x)$ such that $\varphi(\tilde{x})$ evaluates to one. We generalize this idea to Boolean functions $\varphi \in \mathbb{B}(n,m)$, $n,m \in \mathbb{N}$ in the following definition:

Definition 3.7. For a Boolean function $\varphi \in \mathbb{B}(n,m)$ with $n,m \in \mathbb{N}$ and $\emptyset \neq I \subseteq [n]$ we define the projection of the map φ, denoted by $\text{proj}^I(\varphi)$ to be the function $\text{proj}^I(\varphi) : \{0,1\}^{|I|} \to \{0,1\}^m$ in the following way

$$\begin{aligned} \text{proj}^I(\varphi)_i :& \{0,1\}^{|I|} \to \{0,1\}, \\ & x \mapsto \begin{cases} 1 & \text{if } \exists y \in (\text{proj}^I)^{-1}(x) : \varphi_i(y), \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{3.2.1}$$

for any $i \in [m]$.

Note that we can replace the expression $\exists y \in (\mathrm{proj}^I)^{-1}(x) : \varphi_i(y)$ also with a disjunction if convenient:

$$\mathrm{proj}^I(\varphi)_i(x) = \bigvee_{y \in (\mathrm{proj}^I)^{-1}(x)} \varphi_i(y).$$

If we consider an ASTG $G_{\mathrm{async}}(f)$ induced by a Boolean function $f \in \mathbb{B}(n,n)$, and we choose in the above definition $\varphi := \mathrm{cond}^f$, then the condition in (3.2.1) indicates us for each state s in the quotient graph $G_{\mathrm{async}}(f)/\mathrm{proj}^I$ and $i \in I$ if there is a transition $(v, v^{\{i\}})$ somewhere in $G_{\mathrm{async}}(f)$ which is projected to $(s, s^{\{i\}})$. Indeed, it turns out that we can prove, that the graphs $G_{\mathrm{async}}(\mathrm{cond}^{\mathrm{proj}^I(\mathrm{cond}^f)})$ and $G_{\mathrm{async}}(f)/\mathrm{proj}^I$ are exactly the same, if we remove the loops from $G_{\mathrm{async}}(f)/\mathrm{proj}^I$. We also note that according to the remarks at the beginning of Section 3.2 we can identify the states of $G_{\mathrm{async}}(f)/\mathrm{proj}^I$ with the images of proj^I so that it is possible to compare these graphs.

Proposition 3.3. The graphs $G_{\mathrm{async}}(f)/\mathrm{proj}^I$ and $G_{\mathrm{async}}(\mathrm{cond}^{\mathrm{proj}^I(\mathrm{cond}^f)})$ have the same transitions except of possible loops. More precisely, it holds

$$\begin{align} & E_{\mathrm{async}}(\mathrm{cond}^{\mathrm{proj}^I(\mathrm{cond}^f)}) \tag{3.2.2} \\ =& E\big(G_{\mathrm{async}}(f)/\mathrm{proj}^I\big) \setminus \{(v,v) | v \in \{0,1\}^{|I|}\}. \tag{3.2.3} \end{align}$$

Before we prove Proposition 3.3, we give a small example, illustrating the construction of the function $\mathrm{cond}^{\mathrm{proj}^I(\mathrm{cond}^f)}$.

Example 3.5. Consider the ASTG $G_{\mathrm{async}}(f)$ given by

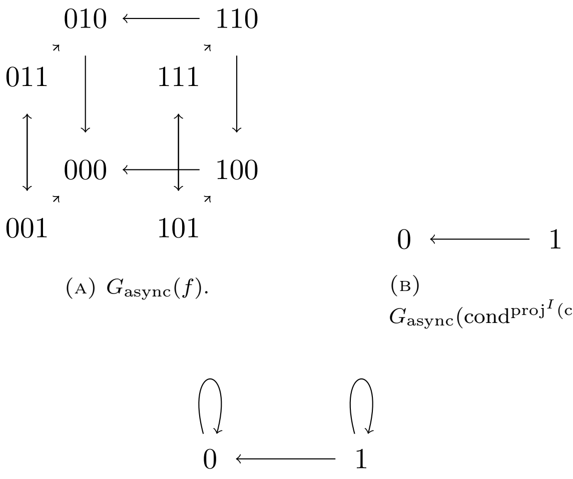
$$\begin{aligned} f :& (x_1, x_2, x_3) \mapsto (x_1 \wedge x_3, \neg x_2 \wedge x_3, 0), \\ \mathrm{cond}^f :& (x_1, x_2, x_3) \mapsto (x_1 \wedge \neg x_3, x_2 \vee x_3, x_3) \end{aligned}$$

depicted in Figure 3.2.4a. Let us consider the ASTG of the projection onto the first component. According to Proposition 3.3 we need to look at the function $\mathrm{proj}^{\{1\}}(\mathrm{cond}^f)$, which is given by

$$\begin{align} \mathrm{proj}^{\{1\}}(\mathrm{cond}^f) : \{0,1\} \to & \{0,1\}, \notag \\ (x_1) \mapsto & \bigvee_{y \in (\mathrm{proj}^I)^{-1}(x_1)} \mathrm{cond}_1^f(y). \tag{3.2.4} \end{align}$$

We can eliminate the quantifier by looking at the DNF (Definition 1.17) of cond_1^f and dropping all the variables corresponding to components in $[n] \setminus I$. More precisely, we can replace the expression (3.2.4) by

$$\begin{aligned} & \bigvee_{y \in (\mathrm{proj}^I)^{-1}(x_1)} \mathrm{cond}_1^f(y) \\ =& \bigvee_{(y_2,y_3) \in \{0,1\}^2} (x_1 \wedge \neg y_3) \\ =& \bigvee_{y_3 \in \{0,1\}^1} (x_1 \wedge \neg y_3) \\ =& x_1, \end{aligned}$$



(A) $G_{\mathrm{async}}(f)$.

(B) $G_{\mathrm{async}}(\mathrm{cond}^{\mathrm{proj}^I(\mathrm{cond}^f)})$.

(C) $G_{\mathrm{async}}(f)/\mathrm{proj}^{\{1\}}$.

FIGURE 3.2.4. Illustration of Example 3.5.

so that we obtain

$$\mathrm{proj}^{\{1\}}(\mathrm{cond}^f) : (x_1) \mapsto (x_1).$$

Consequently, in terms of the update function we obtain $\mathrm{cond}^{\mathrm{proj}^I(\mathrm{cond}^f)} = \mathrm{id} \oplus \mathrm{proj}^{\{1\}}(\mathrm{cond}^f) = x_1 \oplus x_1 = 0$.

Indeed, in Figure 3.2.4c we see the projection of the ASTG $G_{\mathrm{async}}(f)/\mathrm{proj}^{\{1\}}$. It differs from $G_{\mathrm{async}}(\mathrm{cond}^{\mathrm{proj}^I(\mathrm{cond}^f)})$ only in terms of loops.

The observation above motivates the following definition:

Definition 3.8. We define the *ASTG of a projected ASTG* $G_{\mathrm{async}}(f)/\mathrm{proj}^I$ to be the graph $G_{\mathrm{async}}(\mathrm{cond}^{\mathrm{proj}^I(\mathrm{cond}^f)}) = G_{\mathrm{async}}(\mathrm{id} \oplus \mathrm{proj}^I(\mathrm{cond}^f))$.

We conclude this section with the proof of Proposition 3.3.

PROOF. (Proof of Proposition 3.3) Let $\emptyset \neq I \subseteq [n]$ be the set $\{i_1, \ldots, i_{|I|}\}$. For any transition $(v, v^{\{j\}})$ with $v \in \{0,1\}^{|I|}$, $j \in \{1, \ldots, |I|\}$ we have the following equivalence:

$$\begin{aligned}
&(v, v^{\{j\}}) \in E\big(G_{\mathrm{async}}(f)/\mathrm{proj}^I\big) \\
\Leftrightarrow &\exists w \in \{0,1\}^n : \mathrm{proj}^I(w) = v \text{ and } (w, w^{\{i_j\}}) \in E_{\mathrm{async}}(f) \\
\Leftrightarrow &\exists w \in (\mathrm{proj}^I)^{-1}(v) : (w, w^{\{i_j\}}) \in E_{\mathrm{async}}(f) \\
\Leftrightarrow &\exists w \in (\mathrm{proj}^I)^{-1}(v) : \mathrm{cond}^f_{i_j}(w) \\
\Leftrightarrow &\mathrm{proj}^I(\mathrm{cond}^f)_{i_j}(v)
\end{aligned}$$

Using Lemma 3.1 the equality (3.2.2) in Proposition 3.3 follows. □

3.2.1.2. *Some remarks on conditions for equality in Proposition 3.3.* For completeness we give a condition for the equality

$$G_{\mathrm{async}}(f)/\mathrm{proj}^I = G_{\mathrm{async}}(\mathrm{cond}^{\mathrm{proj}^I(\mathrm{cond}^f)}) \tag{3.2.5}$$

in Proposition 3.3. Indeed, Proposition 3.3, implies that (3.2.5) holds iff the graph $G_{\text{async}}(f)/\text{proj}^I$ contains no loops. We show that this is the case if and only if $\text{diff}(E_{\text{async}}(f))$ is a subset of I.

Corollary 3.1. We have:

$$\text{diff}(E_{\text{async}}(f)) \subseteq I \Leftrightarrow G_{\text{async}}(f)/\text{proj}^I \text{ contains no loops.} \tag{3.2.6}$$

Proof. The following equivalences hold

$$\begin{aligned} &G_{\text{async}}(f)/\text{proj}^I \text{ has no loop.}\\ \Leftrightarrow &\text{For all } (v, v^{\{i\}}) \in E_{\text{async}}(f) \text{ it holds } \text{proj}^I(v) \neq \text{proj}^I(v^{\{i\}}).\\ \Leftrightarrow &\text{For all } (v, v^{\{i\}}) \in E_{\text{async}}(f) \text{ it holds } i \in I.\\ \Leftrightarrow &\text{diff}(E_{\text{async}}(f)) \cap I = \emptyset. \end{aligned}$$

□

Using Corollary 3.1 and Proposition 3.3 the following corollary then follows immediately.

Corollary 3.2. We have

$$\text{diff}(E_{\text{async}}(f)) \subseteq I \Leftrightarrow G_{\text{async}}(f)/\text{proj}^I = G_{\text{async}}(\text{cond}^{\text{proj}^I(\text{cond}^f)}).$$

Since the condition in the above corollary is very strong, in most cases there will be loops that we miss in the ASTG $G_{\text{async}}(\text{cond}^{\text{proj}^I(\text{cond}^f)})$.

3.2.1.3. *Relation between projections, subspaces and trap spaces.* It is also worth noticing that trap spaces are closely related to the graph $G_{\text{async}}(\text{cond}^{\text{proj}^I(\text{cond}^f)})$. The I-subspaces in Definition 1.27 can be considered equivalence classes $[s]_{\text{proj}^I}$ of states $s \in \{0,1\}^n$. In fact, it is easy to see that the steady states of the graph $G_{\text{async}}(\text{cond}^{\text{proj}^I(\text{cond}^f)})$ correspond to the I-trap spaces of $G_{\text{async}}(f)$. For this first note:

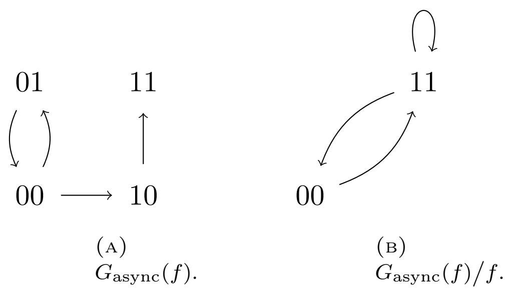
Proposition 3.4. The preimages under the mapping proj^I of the steady states of $G_{\text{async}}(f)/\text{proj}^I$ correspond to the I-subspaces in $G_{\text{async}}(f)$.

Proof. Let p be a steady state of $G_{\text{async}}(f)/\text{proj}^I$ (see Definition 1.9). Since proj^I is a graph homomorphism from $G_{\text{async}}(f)$ onto $G_{\text{async}}(f)/\text{proj}^I$ the preimage $(\text{proj}^I)^{-1}(p)$ is trap set of $G_{\text{async}}(f)$ and therefore a trap space by Definition 1.27.

On the other hand, if $(\text{proj}^I)^{-1}(p)$ is an I-trap space of $G_{\text{async}}(f)$, then by definition for any $s \in (\text{proj}^I)^{-1}(p)$ there are no transitions of the form $(s, s^{\{i\}}) \in E_{\text{async}}(f)$, $i \in I$. Since transitions of the form $(s, s^{\{i\}})$, $i \in [n] \backslash I$ are projected to loops in $G_{\text{async}}(f)/\text{proj}^I$, the state p in $G_{\text{async}}(f)/\text{proj}^I$ has no successors different from itself. By Definition 1.9 it is therefore a steady state of $G_{\text{async}}(f)/\text{proj}^I$. □

Due to Proposition 3.3 we can relate the steady states of $G_{\text{async}}(f)/\text{proj}^I$ and $G_{\text{async}}(\text{cond}^{\text{proj}^I(\text{cond}^f)})$ easily to each other:

Remark 3.1. The steady states of $G_{\text{async}}(f)/\text{proj}^I$ correspond to the steady states of $G_{\text{async}}(\text{cond}^{\text{proj}^I(\text{cond}^f)})$.



(A) $G_{\text{async}}(f)$.

(B) $G_{\text{async}}(f)/f$.

FIGURE 3.2.5. $G_{\text{async}}(f)$ and $G_{\text{async}}(f)/f$ (see Example 3.6).

Taking Proposition 3.4 and Remark 3.1 we obtain the following corollary:

Corollary 3.3. The following equivalence holds

$$\begin{aligned}&\big[p \text{ is a steady state in } G_{\text{async}}(\text{cond}^{\text{proj}^I(\text{cond}^f)})\big]\\ \Leftrightarrow&\big[(\text{proj}^I)^{-1}(p) \text{ is an } I\text{-trapspace in } G_{\text{async}}(f)\big].\end{aligned}$$

3.2.2. The quotient graph with respect to the update function. Another example of an equivalence relation that we will consider here is the identification of states that are mapped to the same image under the update function $f \in \mathbb{B}(n,n)$ of the BN:

$$s \sim_f t :\Leftrightarrow f(s) = f(t)$$

Again, as in the case of projections the equivalence classes of $\sim_f$ can be identified with the preimages of f Instead of $G_{\text{async}}(f)/\sim_f$ we write $G_{\text{async}}(f)/f$. We will investigate families of such quotient graphs in Section 4.2. The quotient graph $G_{\text{async}}(f)/f$ has states corresponding to the equivalence classes induced by $\sim_f$. Since we can identify each equivalence class of a state with its image under f, we can consider the states of $G_{\text{async}}(f)/f$ as states in $\{0,1\}^n$ again.[2] We illustrate this with with two examples:

Example 3.6. Consider the BN depicted in Figure 3.2.5. We can see from the ASTG depicted on the left, that $f^{-1}(11) = \{00, 10, 11\}$ and $f^{-1}(00) = \{01\}$. The corresponding quotient graph consisting of the states 11 and 00 is depicted on the right of Figure 3.2.5.

The size of the quotient graphs $G_{\text{async}}(f)/f$ depends on "the degree of injectivity" of the function f. Consider the following example.

Example 3.7. Consider the Boolean function $f : (x_1, x_2, x_3, x_4) \mapsto (x_2, x_1, x_4 \wedge \neg x_2, \neg x_3)$. Its ASTG is depicted in Figure 3.2.6a. It has 16 states. The quotient graph $G_{\text{async}}(f)/f$ has 12 states. We see for example that there is a transition from the state 0011 to 0010. Since f maps 0011 to 0010 and 0010 to 0000, the quotient graph $G_{\text{async}}(f)/f$ contains a transition 0010 → 0000. It is depicted in Figure 3.2.6b.

Since the interaction graphs of Boolean models of GRNs are in general very sparse, the function f can not vary along many dimensions. Therefore,

[2] But note that not necessarily all states in $\{0,1\}^n$ are covered by f.

for larger networks the quotient graphs $G_{\text{async}}(f)/f$ should be considerable smaller than $G_{\text{async}}(f)$. To give some intuition and to point to future possible applications we list some simple properties of the graph $G_{\text{async}}(f)/f$.

Proposition 3.5. If s is a steady state of $G_{\text{async}}(f)/f$ then s is a steady state of $G_{\text{async}}(f)$ and $f^{-1}(s) \subseteq \text{StrongBasin}(\{s\})$. More precisely it holds:
$\underline{\textbf{A)}}$ For any $t \in f^{-1}(s)$ all sufficiently long directed paths in $G_{\text{async}}(f)$ starting in t will end in s and each path consists of $|\text{diff}(t, s)|$ states not equal to s.
$\underline{\textbf{B)}}$ If $t \in f^{-1}(s)$ then $\{t^A | A \subseteq \text{diff}(t, s)\} \subseteq f^{-1}(s)$.

Proof. **A)**: Let t be any state in $f^{-1}(s)$ with $s \neq t$ and let (t, u) be any transition in $E_{\text{async}}(f)$. Since s is a steady state in $G_{\text{async}}(f)/f$ the set $f^{-1}(s)$ is a trap set in $G_{\text{async}}(f)$ and therefore also $u \in f^{-1}(s)$. It holds $d(u, s) < d(t, s)$, where $d(\cdot, \cdot)$ denotes the Hamming distance (see Definition 1.14). By applying the same argument iteratively to u, we obtain that every sufficiently long path starting in t will eventually end in s.
B): Let $A = \{a_1, \ldots a_k\}$. Since the inclusion $A \subseteq \text{diff}(t, s)$ holds there is a transition $(t, t^{\{a_i\}})$ in $E_{\text{async}}(f)$ and since $f^{-1}(s)$ is a trap set in $G_{\text{async}}(f)$ it holds $f(t^{\{a_i\}}) = s$. Using the same argument iteratively we obtain that $t^A \in f^{-1}(s)$. □

Note also that as an immediate consequence of Proposition 3.5 we obtain that the number of steady states of any quotient graph $G_{\text{async}}(f)/f$ of an ASTG $G_{\text{async}}(f)$ is smaller or equal to the number of steady states in the original ASTG.

Corollary 3.4. Let $f \in \mathbb{B}(n, n)$ be the update function of an ASTG $G_{\text{async}}(f)$. The number of steady states in $G_{\text{async}}(f)/f$ is smaller or equal than the number of steady states in $G_{\text{async}}(f)$. For each steady state s in $G_{\text{async}}(f)/f$ there is exactly one steady state $t \in f^{-1}(s)$ in the ASTG $G_{\text{async}}(f)$.

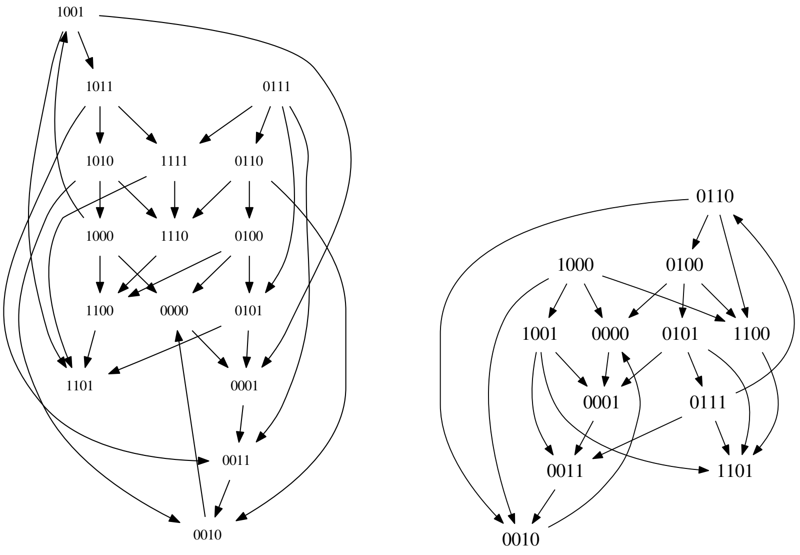
More generally, we have a similar statement about subspaces in $G_{\text{async}}(f)/f$. In fact, if $[s]_{\text{proj}^I}$ is a trap set in $G_{\text{async}}(f)/f$ then all states in its preimage $f^{-1}([s]_{\text{proj}^I})$ have a path into $f^{-1}([s]_{\text{proj}^I}) \cap [s]_{\text{proj}^I}$. We formalize this in the following proposition.

Proposition 3.6. If for some non-empty set $I \subseteq [n]$ the set $[s]_{\text{proj}^I} \cap f(\{0, 1\}^n)$, $s \in \{0, 1\}^n$ is a non-empty trap set in $G_{\text{async}}(f)/f$, then the sets $f^{-1}([s]_{\text{proj}^I})$ and $f^{-1}([s]_{\text{proj}^I}) \cap [s]_{\text{proj}^I}$ are non-empty trap sets in $G_{\text{async}}(f)$. Furthermore, $f^{-1}([s]_{\text{proj}^I})$ is in the strong basin of attraction of $f^{-1}([s]_{\text{proj}^I}) \cap [s]_{\text{proj}^I}$, i.e.

$$f^{-1}([s]_{\text{proj}^I}) \subseteq \text{StrongBasin}(f^{-1}([s]_{\text{proj}^I}) \cap [s]_{\text{proj}^I}).$$

.

Proof. The set $[s]_{\text{proj}^I} \cap f(\{0, 1\}^n)$ is a trap set in $G_{\text{async}}(f)/f$. Therefore, its preimage $f^{-1}\big([s]_{\text{proj}^I} \cap f(\{0, 1\}^n)\big) = f^{-1}([s]_{\text{proj}^I})$ is a non-empty trap set in $G_{\text{async}}(f)$. For all $i \in I$ all states $t \in f^{-1}([s]_{\text{proj}^I})$ are mapped to s_i under f_i.
Due to the inclusion $[s]_{\text{proj}^I} \cap f^{-1}([s]_{\text{proj}^I}) \subseteq f^{-1}([s]_{\text{proj}^I})$ the same holds



(A) $G_{\text{async}}(f)$ in Example 3.7.

(B) The graph $G_{\text{async}}(f)/f$ (See Example 3.7). Loops are omitted. We see that the graph $G_{\text{async}}(f)/f$ has the transition 0010 → 0000.

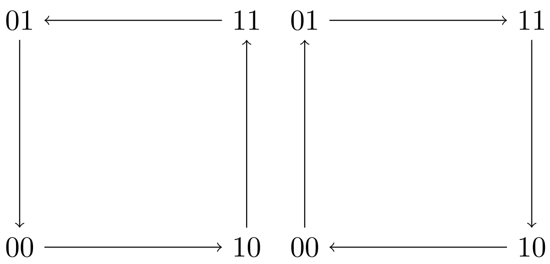
FIGURE 3.2.6. Illustration of the ASTG in Example 3.7 and its quotient graph.

true for the states in $[s]_{\text{proj}^I} \cap f^{-1}([s]_{\text{proj}^I})$. This shows that also $[s]_{\text{proj}^I} \cap f^{-1}([s]_{\text{proj}^I})$ is a trap set in $G_{\text{async}}(f)$.
Furthermore, with the same argument it follows that from each
$t \in f^{-1}([s]_{\text{proj}^I})$ there is a path in $G_{\text{async}}(f)$ leading into $[s]_{\text{proj}^I} \cap f^{-1}([s]_{\text{proj}^I})$. Consequently, the set $[s]_{\text{proj}^I} \cap f^{-1}([s]_{\text{proj}^I})$ is not empty and by Definition 1.10 the inclusion $f^{-1}([s]_{\text{proj}^I}) \subseteq \text{StrongBasin}([s]_{\text{proj}^I})$ holds. □

Note that if we choose in Proposition 3.6 a trap space $[s]_{\text{proj}^I}$ consisting of just one element, i.e. a steady state, then we arrive at Proposition 3.5.

The above points give some examples of the type of knowledge we can obtain from the quotient graphs about the dynamics of the network. We demonstrate this with our running example.

Example 3.8. Continuing Example 3.7 we see in Figure 3.2.6b that the set $00{*}{*}$ is a trap set in $G_{\text{async}}(f)/f$. It holds $f^{-1}(00{*}{*}) = 00{*}{*}$, which we can easily check in Figure 3.2.6a is a trap space in $G_{\text{async}}(f)$. Furthermore 1101 is a steady state in $G_{\text{async}}(f)/f$. We have $f^{-1}(1101) = 110{*}$. According to Proposition 3.5 1101 is a steady state in $G_{\text{async}}(f)$ and the set $110{*}$ is in its strong basin of attraction. Again, in Figure 3.2.6a we can see that this is indeed the case.



(A) The ASTG $G_{\text{async}}(f)$.

(B) The ASTG $G_{\text{async}}(g)$.

FIGURE 3.3.1. Illustration of Example 3.9.

3.3. Inverting transitions

As investigated in Fuey et al. [2017] sometimes it is useful to invert transitions of an ASTG. This can be easily achieved by the following proposition.

Proposition 3.7. For any $f \in \mathbb{B}(n, n)$ it holds

$$(v, w) \in E_{\text{async}}(f) \Leftrightarrow (w, v) \in E_{\text{async}}(g)$$

with $g_i(x) := \neg f_i(x^{\{i\}})$ for all $x \in \{0,1\}^n$, $i \in [n]$.

PROOF. We have the equivalences:

$$\begin{aligned} &(v, v^{\{i\}}) \in E_{\text{async}}(g) \\ \Leftrightarrow &\text{cond}_i^g(v) \\ \Leftrightarrow &\text{cond}_i^f(v^{\{i\}}) \\ \Leftrightarrow &(v^{\{i\}}, v) \in E_{\text{async}}(f) \end{aligned}$$

Furthermore, it holds $g_i(x) := x_i \oplus \text{cond}_i^f(x^{\{i\}}) = x_i \oplus x_i^{\{i\}} \oplus f_i(x^{\{i\}}) = \neg f_i(x^{\{i\}})$. □

We give a small example:

Example 3.9. Consider the Boolean function $f : (x_1, x_2) \mapsto (\neg x_2, x_1)$. Its ASTG is depicted in Figure 3.3.1a. According to Proposition 3.7 we have for $g : (x_1, x_2) \mapsto (x_2, \neg x_1)$ the relation $(v, w) \in E_{\text{async}}(f) \Leftrightarrow (w, v) \in E_{\text{async}}(g)$. Indeed looking on the ASTGs depicted in Figure 3.3.1 we see that this is the case.

3.4. Discussion

We discussed several ways to perform set operations on ASTGs in terms of the update functions of these ASTGs. This included simple set operations such as the union or intersection of ASTGs. Then we considered two types of quotient graphs, which will be used in the following chapters. If possible we tried to formulate these operations without considering the graph representation. This has the advantage that the state transition graph does not need to be constructed explicitly.

There are different aspects not mentioned here, that are also of interest. For example we restricted ourselves to the asynchronous update scheme. It would be interesting to consider these operations from a more general point of view and see how they can be used in arbitrary BNs. We also did not consider any relations to the interaction graphs. We will come back to this question in the next chapter for one type of quotient graph considered here.

CHAPTER 4

Model pools and their graph representation

Contents

4.1. The qualitative state transition graph of a continuous monotonic model pool 64

4.1.1. A graph representation of a continuous monotonic model pool 64

4.1.2. Proof of the main result 69

4.1.3. Boolean reformulation of the QSTG 74

4.2. The qualitative state transition graph of a Boolean monotonic model pool 75

4.2.1. A graph representation of a Boolean monotonic model pool 76

4.2.2. Proof of the main result 79

4.2.3. Some further remarks: other Boolean model pools 84

4.2.4. $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ is a subgraph of $G_{\mathrm{QSTG}}(\Sigma)$. 85

4.3. The skeleton of the qualitative state transition graphs 86

4.3.1. The skeleton of the graph $G_{\mathrm{QSTG}}(\Sigma)$ 87

4.3.2. The skeleton of the graph $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ 88

4.4. Reconstruction of the interaction graph 93

4.4.1. There is a one-to-one correspondence between Σ, $G_{\mathrm{async}}(f^{\Sigma})$, $G_{\mathrm{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$. 94

4.4.2. A network inference algorithm 96

4.4.3. Numerical experiments 97

4.5. Discussion 98

In this chapter we describe different methods for investigating sets of models. Mathematical modeling in systems biology results often in a potentially very large set of models which agree with given data. Therefore, instead of investigating a specific model, a set of possible models has to be investigated. In this chapter we consider the scenario where all these models have a common structure but their dynamics are allowed to vary. We are interested in deducing restrictions on the dynamics from structural descriptions of the network. This constitutes a scenario of particular interest in applications, where interaction information is usually more readily available than details on the processing logic of multiple influences on a target component. This scenario arises in different modeling frameworks Kuipers [1984], Kaufman et al. [2007], Remy et al. [2008], Soulé [2003], Thomas [1981], Thomas and Kaufman [2001b].

The other aspect of the results here is that they relate interaction graphs, Boolean models and ODE models. In Chapter 2 we looked on very specific aspects of the dynamics of models (steady states and trap spaces). Our results here in Chapter 4 will take a different more general point of view comparing complete model sets arising from different modeling formalisms. We will compare two methods of abstraction of a family of ODEs and a family of BNs and show that the result of both methods is almost identical.

The structure of the network is represented as an interaction graph. The mathematical interpretation of the interaction graph depends on the chosen modeling framework.

In Section 4.1 we will consider families of ODE systems. We revisit the theory of *qualitative differential equations* (QDE) Eisenack [2006]. Here, a signed interaction graph is interpreted as the sign structure Σ of a Jacobian matrix of an ODE system. All ODE systems which are consistent with a given interaction graph are collected in a so-called *continuous monotonic model pool* $\mathscr{M}(\Sigma)$. For this model pool, a *qualitative state transition graph* (QSTG) $G_{\text{QSTG}}(\Sigma)$ can be constructed whose states represent derivative signs of the system components and transitions indicate possible changes in the derivative over time. This graph can then be used to describe the behavior of the model pool. At the end of Section 4.1 we will reformulate this result in such a way that the graph $G_{\text{QSTG}}(\Sigma)$ can be considered as a Boolean state transition graph, i.e. a graph whose nodes are Boolean vectors in $\{0,1\}^n$. This reformulation will be used in Section 4.3 and subsequent chapters.

In Section 4.2 we consider families of Boolean models. In this case the number of models that agree with a given interaction graph is finite. However, analysis of every single model in the set is costly. Exploiting formal verification techniques allows to investigate the behavior of large numbers of models in this context Streck et al. [2016], Videla et al. [2017]. A different approach aims at avoiding enumeration and explicit analysis of every model in the set by deriving properties directly from the given constraints, e.g., inferring dynamical information from coinciding structural characteristics of all models Thomas and Kaufman [2001a], Remy et al. [2008], Thomas [1981], Thomas and Kaufman [2001b]. Here, we adopt the latter approach for sets of BNs consistent with a given signed interaction graph Σ.

Analogous to Section 4.1, we show that a state transition graph $G^{\mathbb{B}}_{\text{QSTG}}$ captures behavioral restrictions on the dynamics exhibited by the models in the model pool. More specifically, for each Boolean function f consistent with a given interaction graph, we are interested in the ASTG capturing the dynamics of the model. We show that while the ASTG cannot be related directly to $G^{\mathbb{B}}_{\text{QSTG}}$ this becomes possible for a quotient graph derived from the ASTG by identifying system states with the same image under f. This quotient graph needs to be a subgraph of $G^{\mathbb{B}}_{\text{QSTG}}$. Consequently, analysis of $G^{\mathbb{B}}_{\text{QSTG}}$ allows to infer reachability constraints valid for all models consistent with the interaction graph. In particular, universal statements about trap sets and attractors become possible. The results of this section are partially based on Schwieger and Siebert [2017, 2018].

Finally, in Section 4.3 we compare the graphs $G_{\text{QSTG}}(\Sigma)$ of Section 4.1 and $G^{\mathbb{B}}_{\text{QSTG}}$ of Section 4.2. It turns out that $G^{\mathbb{B}}_{\text{QSTG}}$ can be considered a subgraph of $G_{\text{QSTG}}(\Sigma)$ when identifying the states of both graphs in a reasonable way with each other. Furthermore, the two graphs have the same reachability properties. Our final result in Section 4.3 will be the construction of an ASTG $G_{\text{async}}(f^{\Sigma})$, which will be a subgraph of $G^{\mathbb{B}}_{\text{QSTG}}$ and $G_{\text{QSTG}}(\Sigma)$. We will call this graph the *skeleton of the qualitative state*

$$\begin{tikzcd}
\mathscr{M}(\Sigma, X) \arrow[rr, dashed, "\text{Abstracting the reasonable solutions of the ODEs in } \mathscr{M}(\Sigma, X)"] & & \tilde{G}_{\mathrm{QSTG}}(\Sigma) \\
\text{Sign matrix } \Sigma \arrow[u, dashed] \arrow[urr, dashed, "\text{Construction via Prop. 4.1}"] & &
\end{tikzcd}$$

FIGURE 4.1.1. Sketch of the construction of the graph $\tilde{G}_{\mathrm{QSTG}}(\Sigma)$ for a sign-matrix $\Sigma \in \{-1, 0, 1\}^{n \times n}$. Instead of solving the potentially infinite ODEs in $\mathscr{M}(\Sigma, X)$ we can construct the graph $\tilde{G}_{\mathrm{QSTG}}(\Sigma)$ with Proposition 4.1 directly from the sign matrix Σ.

transition graphs, since we can show that it contains already all information about reachability of both graphs.

4.1. The qualitative state transition graph of a continuous monotonic model pool

This section is based on [Eisenack, 2006, Section 2.2.1]. We will start by reviewing these results. This includes the introduction of the continuous monotonic model pool, i.e. the set of models we are investigating. Then abstractions of the solutions of the ODEs in this model pool are introduced. These abstractions in turn are used to define a graph capturing possible behavior. Such a graph would not be useful, if it would need to be explicitly generated from solutions of ODE-system. Therefore, the most important result of this section is Proposition 4.1, which gives an equivalent description of this graph which only depends on the description of the model pool. Finally in Section 4.1.3 we give a Boolean reformulation (see also Schwieger and Siebert [2017]) of this graph. The idea of this section is captured in Figure 4.1.1.

4.1.1. A graph representation of a continuous monotonic model pool. In this section we analyze families of ODE models $\dot{x} = f(x)$ which share some qualitative properties. These qualitative properties are sign constraints on the Jacobian matrix of the right hand sides of the ODE-System. Instead of the solutions $x(\cdot)$ of the ODE systems, so-called abstractions are considered. The abstractions considered here are intervals on $\mathbb{R}_{\geq 0}$ on which the component functions of $x(\cdot)$ are either increasing or decreasing.

Definition 4.1. Let $x : J \to \mathbb{R}$ be a real valued function defined on an interval $J \subset \mathbb{R}$.
1) x is increasing on J if $\forall t_1, t_2 \in J : t_1 < t_2 \Rightarrow x(t_1) < x(t_2)$ holds.

2) x is decreasing on J if $\forall t_1, t_2 \in J : t_1 < t_2 \Rightarrow x(t_1) > x(t_2)$ holds.
3) x is strictly monotonic on J if it is either increasing or decreasing.

Due to the monotonicity theorem we can characterize these intervals by the derivatives of the solutions:

Theorem 4.1 (Monotonicity Theorem)**.** Let $x : J \to \mathbb{R}$ be a continuous real valued function defined on an interval $J \subset \mathbb{R}$, which is differentiable inside J.
1) If $\dot{x}(t) > 0$ for all t in the interval J, then x is increasing on J.
2) If $\dot{x}(t) < 0$ for all t in the interval J, then x is decreasing on J.

Therefore, we can consider abstractions as sequences of sign vectors of the derivatives of the solutions. A state transition graph on the set of sign vectors can be constructed based on the sign matrix, which captures restrictions on the behavior of the solutions. This graph will be the focus of our investigation in this section.

We define a set of models – referred to as model pool – to be a set of ODE systems whose corresponding Jacobian matrices share a sign structure. For this purpose we define:

Definition 4.2. The map $\operatorname{sign}(\cdot) : \mathbb{R} \to \{-1, 0, 1\}$ is defined as follows:

$$\operatorname{sign}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}.$$

We extend this mapping componentwise to vectors and matrices over $\mathbb{R}$.

We use this notation to define the continuous monotonic model pool to be the set of r.h.s. functions of ODE systems whose Jacobian matrix has a specific sign structure.

Definition 4.3 ([Eisenack, 2006, p. 22])**.** For a given $n \times n$ matrix of signs $\Sigma = (\sigma_{i,j})_{i,j \in [n]}$, $\sigma_{i,j} \in \{-1, 0, 1\}$ and a state space $X \subseteq \mathbb{R}^n$ we define the *continuous monotonic model pool*

$$\mathscr{M}(\Sigma, X) := \{f \in C^1(X, \mathbb{R}^n) \mid \forall x \in X : \operatorname{sign}(J_f(x)) = \Sigma\},$$

where J_f denotes the Jacobian of f.
We call a function $x \in C^1([0,T], \mathbb{R}^n)$, $T \in (0, \infty]$, *reasonable*, if there is only a finite set of points t with $\dot{x}(t) = 0$ in any bounded interval and for all local extrema of x_i, $i \in [n]$ the second derivative does not vanish. We define the *space of admissible trajectories* by

$$\mathscr{E} := \big\{x \in C^1([0,T], X) | x \text{ is reasonable}\big\}.$$

The *solution set* $S_{\mathscr{M}(\Sigma,X)}$ for an initial value problem with $x(0) = x_0 \in X$ contains all reasonable solutions of corresponding ODE systems whose right hand side function is contained in the model pool $\mathscr{M}(\Sigma, X)$, i.e.,

$$S_{\mathscr{M}(\Sigma,X)} := \{x \in \mathscr{E} | \exists f \in \mathscr{M}(\Sigma, X), x_0 \in X \text{ s.t. } \dot{x} = f(x), x(0) = x_0\}.$$

In summary, we consider the set of ODE systems with a specific sign structure for investigating a certain class of solutions – reasonable solutions – of these ODE systems. For conciseness, we will identify an ODE with its

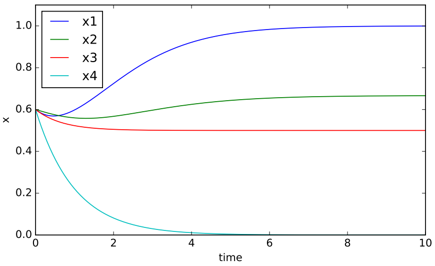


FIGURE 4.1.2. Trajectories of a solution of an ODE in the continuous monotonic model pool. Its abstraction is given by $(-1,-1,-1,-1) \to (1,-1,-1,-1) \to (1,1,-1,-1)$. The ODE was parametrized with $x_0 = (0.6, 0.6, 0.6, 0.6)$, $d = (1,1,1,1)$, $\theta = (0.5, 0.5, 0.5, 0.5)$ and $k = (1,1,1,1)$.

right hand side function, talking about ODEs as elements of the continuous monotonic model pool.

Remark 4.1. Note that our definition is slightly different than the definition given in [Eisenack, 2006, p. 22]. We add here the constraint that the second derivatives do not vanish at local extrema. We will see later that this constraint is actually necessary for the results in [Eisenack, 2006, Prop. 2].

We illustrate the notions on our running example.

Example 4.1. Consider all solutions of ODE systems $\dot{x} = f(x)$ with $f \in C^1([0,1]^4, \mathbb{R}^4)$ having a Jacobian matrix with the sign structure

$$\Sigma = \begin{pmatrix} -1 & 0 & 0 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & -1 & -1 \end{pmatrix}.$$

They constitute a model pool denoted by $\mathscr{M}(\Sigma)$. As an example for elements of this model pool, we construct now a function $f \in \mathscr{M}(\Sigma)$:

$$\begin{aligned} \dot{x} &= \tilde{f}(x) - x, \\ x(0) &= x_0. \end{aligned}$$

with $\tilde{f} : [0,1]^4 \to [0,1]^4$ given by

$$\tilde{f}(x) = \left(1 - \tfrac{x_4}{x_4+0.5}, \quad \tfrac{x_1}{x_1+0.5}, \quad \tfrac{x_2}{x_2+0.5} \cdot \left(1 - \tfrac{x_4}{x_4+0.5}\right), \quad 1 - \tfrac{x_3}{x_3+0.5}\right)^t.$$

It can easily be checked that the map f is in the model pool $\mathscr{M}(\Sigma)$.[1] Figure 4.1.2 shows the solution of the ODE-System, if we choose $x_0 = \begin{pmatrix} 0.6 & 0.6 & 0.6 & 0.6 \end{pmatrix}^t$, and illustrates that it is a reasonable function, thus belonging to the solution set.

We are now looking for qualitative features of solutions in order to find properties common to all ODEs in the continuous monotonic model pool.

[1] In fact the function $\tilde{f}$ can be seen as a Boolean function $f \mapsto \begin{pmatrix} 1 - x_4 & x_1 & x_2 \wedge \neg x_4 & \neg x_3 \end{pmatrix}^t$ converted into a continuous function with the methods considered in Chapter 2.

The idea is to obtain a rough description of solution trajectories by keeping track of the sign changes in the derivative. In general, to each reasonable solution $x : [0, T] \to X$, T finite, we can assign a unique ordered, maximal sequence $(t_j)_{j \in \{0,\ldots,M\}}$, $t_j \in [0, T]$, with $t_0 = 0$, $t_M = T$ and $t_j \in (0, T)$ with the vector $\dot{x}(t_j)$ having a zero entry for $j \in [M - 1]$ indicating sign jumps of the trajectory.[2] If T is infinite we can define a similar sequence not ending in T. In the interval between the points in the sequence the sign of the solution derivative is then constant. We are now interested in the sequence of those signs. Formally, we get the following definition.

Definition 4.4 ([Eisenack, 2006, p. 23])**.** For a solution $x \in \mathscr{E}$, consider the ordered sequence (t_j) in $[0, T]$ consisting of 0 and all boundary points of the closure of all sets $\{t \in [0, T] \big| \operatorname{sign}(\dot{x}(t)) = v\}$ with $v \in \{-1, 1\}^n$. A sequence $(\tau_j) \subset [0, T]$ with $\tau_j \in (t_j, t_{j+1})$ and $\dot{x}_i(\tau_j) \neq 0$ for all $i \in [n]$ gives rise to the sequence $\tilde{x} = (\tilde{x}_j) := (\operatorname{sign}(\dot{x}(\tau_j))$ which is called *abstraction* of $x(\cdot)$.[3] We denote the *set of abstractions* of the solutions of the model pool $\mathscr{M}(\Sigma)$ by

$$\tilde{S}_{\mathscr{M}(\Sigma)} := \big\{\tilde{x} \mid \tilde{x} \text{ is the abstraction of } x(\cdot) \text{ for some } x \in S_{\mathscr{M}(\Sigma)}\big\}.$$

The sign vectors in the above definitions are elements of $\{-1, 1\}^n$. Subsequent elements in an abstraction are always different from each other, since the sequence (t_j) is chosen from the closure of the sets $\{t \in [0, T] \big| \operatorname{sign}(\dot{x}(t)) = v\}$ and the function x is reasonable. This has the effect that the sequence (t_j) does not contain points, where the derivates of all components vanish, but subsequently take the same sign as before (e.g. saddle points). To illustrate the notion, we extract the abstraction for a solution of the running example.

Example 4.2. Consider the solution of the ODE depicted in Figure 4.1.2. Its abstraction is given by the sequence $((-1, -1, -1, -1), (1, -1, -1, -1), (1, 1, -1, -1))$, since at the beginning of the trajectory all components are decreasing. Then the first component starts increasing followed by the second one.

Based on the abstractions a so-called qualitative state transition graph can now be constructed. The states correspond to the signs of $\dot{x}(\cdot)$ and transitions indicate subsequent sign vectors in some abstraction.

Definition 4.5 ([Eisenack, 2006, p. 23])**.** The *qualitative state transition graph (QSTG)* $\tilde{G}_{QSTG}(\Sigma)$ *of the continuous monotonic model pool* is defined by the set of nodes

$$\tilde{V}_{\text{QSTG}}(\Sigma) := \{-1, 1\}^n,$$

called *qualitative states*, and the set of edges

$$\tilde{E}_{\text{QSTG}}(\Sigma) := \big\{(v, w) | \exists \tilde{x} \in \tilde{S}_{\mathscr{M}(\Sigma)}, j \in \mathbb{N} : \tilde{x}_j = v \text{ and } (\tilde{x}_j = w \text{ or } \tilde{x}_{j+1} = w)\big\},$$

called *qualitative transitions*.

[2] $[M - 1]$ denotes the set $\{1, \ldots, M - 1\}$ (see Section 1.1).

[3] The additional assumption, not made in [Eisenack, 2006, p. 23], that none of the components of $\dot{x}(\tau_j)$ vanishes, is necessary, since in principle there could be for example saddle points on the interval (t_j, t_{j+1}), which are not included in the sequence of boundary points (t_j). Note also, that since x is assumed to be reasonable, it is always possible to choose such points $\tau_j \in (t_j, t_{j+1})$.

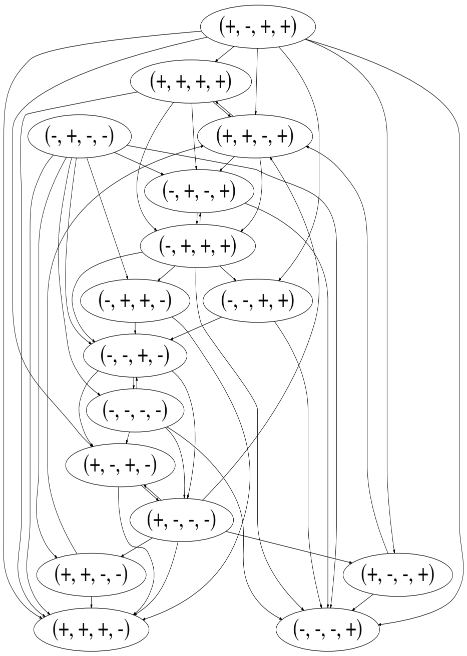


FIGURE 4.1.3. The graph $\tilde{G}_{\mathrm{QSTG}}(\Sigma)$ of the model pool in the running example constructed with Proposition 4.1. We denote 1 with the symbol + and -1 with the symbol $-$. Self-loops are omitted from $\tilde{G}_{\mathrm{QSTG}}(\Sigma)$.

In the following, we indicate the transitions with $\to$. It is easy to see that for each qualitative state v there is a self-loop (v, v) in $\tilde{G}_{QSTG}(\Sigma)$.[4]

Example 4.3. The QSTG of our running example is depicted in Figure 4.1.3. We see that we can find the path $(-1, -1, -1, -1) \to (1, -1, -1, -1)$ $\to (1, 1, -1, -1)$ of Example 4.2 in this graph.

Naturally, this graph would not be very helpful in applications if we would need to solve all ODEs in the model pool to construct it. The following proposition constitutes a different approach. It basically says that a change in the sign of a component i must be caused by a consistent dependency on a component j in the right hand side function f, as captured in the i, j-th entry $\sigma_{i,j}$ of the sign matrix Σ.

Proposition 4.1 ([Eisenack, 2006, p. 25])**.** Let $v, w \in \{-1, 1\}^n = \tilde{V}_{\mathrm{QSTG}}(\Sigma)$. Then, $(v, w) \in \tilde{E}_{\mathrm{QSTG}}(\Sigma)$ iff

$$\forall i \in \mathrm{diff}(v, w) \exists j \in \mathrm{comm}(v, w) : w_i \cdot v_j = \sigma_{i,j}, \tag{4.1.1}$$

where $\mathrm{diff}(\cdot, \cdot)$ and $\mathrm{comm}(\cdot, \cdot)$ were defined in Definition 1.2.

The proof of Proposition 4.1 will be given in a moment (Section 4.1.2). To conclude this section, we note that there is no one-to-one correspondence between the set of QSTGs and the set of sign matrices. It is possible to

[4] At this point we differ slightly from the definition of the QSTG in [Eisenack, 2006, p. 24], where the QSTG contains no self-loops.

change elements on the diagonal of Σ without changing the graph $\tilde{G}_{\mathrm{QSTG}}(\Sigma)$. This is due to the fact that the sets $\mathrm{diff}(v, w)$ and $\mathrm{comm}(v, w)$ are disjoint and thus the diagonal elements do not play a role in (4.1.1). Consequently, the set of transitions does not change when changing the diagonal of Σ.

Remark 4.2. Let Σ be a sign matrix. Then the following equality holds

$$\tilde{G}_{\mathrm{QSTG}}(\Sigma) = \tilde{G}_{\mathrm{QSTG}}(\Sigma - D),$$

where D is any diagonal matrix with entries in $\{-1, 0, 1\}$.

However, we will see later, that if we identify sign matrices, which differ only on elements on the diagonal, the correspondence becomes one-to-one (see Proposition 4.9 and Corollary 4.6 in Section 4.4.1).

4.1.2. Proof of the main result. We prove now Proposition 4.1. The reader not interested in the technical details is advised to skip this section. The proof is according to [Eisenack, 2006, p. 25] with a few modifications. Assume for this section that Σ is a $n \times n$ sign matrix, i.e. $\Sigma = (\sigma_{i,j})_{i,j \in [n]} \in \{-1, 0, 1\}^{n \times n}$. The proof contains two directions. We start by proving that if the condition for a transition in Proposition 4.1 is satisfied, there exists indeed an ODE-system in the model pool with a reasonable solution whose abstraction contains somewhere this transition. To do so, we start by constructing an ODE-system with r.h.s. $f(x) = Ax + c$ for some suitable chosen matrix A and vector c.

Also note that for both directions of the proof we can focus on the transitions of the form (v, w), $w \neq v$, since for any self-loop Condition (4.1.1) is trivially satisfied. It is easy to see that for each $v \in \tilde{V}_{\mathrm{QSTG}}(\Sigma)$ there is a transition $(v, v) \in \tilde{G}_{\mathrm{QSTG}}(\Sigma)$, since no assumptions about the second derivative are made for the solutions in $S_{\mathscr{M}(\Sigma)}$ [Eisenack, 2006, p. 24].

Lemma 4.1. Let (v, w), $v, w \in \{-1, 1\}^n$, $v \neq w$ be a tuple satisfying Condition (4.1.1) and k be a mapping $k : \mathrm{diff}(v, w) \to \mathrm{comm}(v, w)$ such that for all $i \in \mathrm{diff}(v, w)$ it holds

$$w_i \cdot v_{k(i)} = \sigma_{i,k(i)}.$$

Then for the matrix $A := A^{(v,w),k}$ defined by

$$a_{i,j} = \begin{cases} n \cdot w_i \cdot v_j & \text{if } i \in \mathrm{diff}(v, w) \text{ and } j = k(i), \\ \sigma_{i,j} & \text{otherwise.} \end{cases}$$

it holds $\mathrm{sign}(A) = \Sigma$.

Proof. We only need to show that for $i \in \mathrm{diff}(v, w)$ and $j = k(i)$ it holds $\mathrm{sign}(a_{i,j}) = \sigma_{i,j}$. But by definition of k it holds $w_i \cdot v_{k(i)} = \sigma_{i,k(i)}$. Therefore, we have $a_{i,j} = n \cdot w_i \cdot v_{k(i)} = n \cdot \sigma_{i,k(i)}$ and consequently it holds $\mathrm{sign}(a_{i,j}) = \mathrm{sign}(n \cdot \sigma_{i,k(i)}) = \sigma_{i,k(i)}$. □

The mapping k in the above Lemma chooses an index such that the corresponding sign change in the i-th component could be caused by the $k(i)$-th component. By choosing the entries $a_{i,k(i)}$ sufficiently large, we can guarantee that the corresponding sign change in the i-th component will indeed appear in the abstraction of a solution of the corresponding ODE-system. We make this now precise. For this purpose, we use the matrix A from the above lemma to construct a suitable ODE-system:

Lemma 4.2. Assume for $v, w \in \{-1,1\}^n$ with $v \neq w$ that Condition (4.1.1) is satisfied. Then it holds:

$$\begin{aligned} &\exists f \in \mathscr{M}(\Sigma), x' \in X, t_0 < t_1 < t_2 \in \mathbb{R}_+ \\ &\qquad \text{and a solution } x \in S_{\mathscr{M}(\Sigma,X)} \text{ with } \dot{x} = f(x), x(0) = x' \text{ and} \\ &\qquad \forall t \in [t_0, t_1) : \operatorname{sign}(\dot{x}(t)) = v, \\ &\qquad \forall t \in (t_1, t_2] : \operatorname{sign}(\dot{x}(t)) = w \end{aligned} \tag{4.1.2}$$

.

Proof. We construct appropriate f,x',t_0, t_1 and t_2. Choose $c \in \mathbb{R}^n$ such that

$$c_i = \begin{cases} 0 & \text{if } i \in \operatorname{diff}(v,w), \\ v_i & \text{if } i \in \operatorname{comm}(v,w), \end{cases}$$

holds. Let $k : \operatorname{diff}(v,w) \to \operatorname{comm}(v,w)$ be a mapping and A be a matrix $A = A^{(v,w),k} = (a_{i,j}) \in \mathbb{R}^{n \times n}$ in the sense of Lemma 4.1. Now we define

$$f(x) := Ax + c.$$

Again due to Lemma 4.1 it holds $f \in \mathscr{M}(\Sigma)$, since $J_f(\cdot) = A$. We choose an arbitrary $t_1 \in \mathbb{R}_{>0}$ and an open interval J_0 with $t_1 \in J_0$ such that $x(\cdot) : J_0 \to \mathbb{R}^n$ is a solution of the ODE

$$\begin{aligned} \dot{x} &= f(x), \\ x(t_1) &= 0. \end{aligned}$$

Next, we show that for each $i \in \operatorname{comm}(v,w)$ we can find an interval around t_1 such that the sign of $\dot{x}_i$ on that interval is v_i.[5] And for each $i \in \operatorname{diff}(v,w)$ we can find an interval around t_1 such that the sign of the second derivative $\ddot{x}_i$ is w_i. Then we will take the intersection of these n intervals and the interval J_0, and show that the abstractions on this interval correspond to the transition $v \to w$ described in (4.1.2).

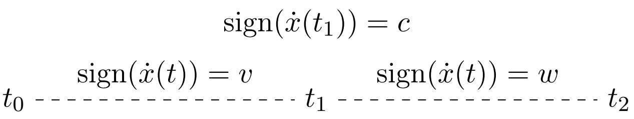


Figure 4.1.4. Illustration of the proof of Lemma 4.2.

Observation 1: For $i \in \operatorname{comm}(v,w)$ exists an interval J_i with $t_1 \in J_i$ such that $\forall t \in J_0 : \operatorname{sign}(\dot{x}_i(t)) = v_i$ holds.
Proof: It holds $\dot{x}(t_1) = \underbrace{A \cdot x(t_1)}_{=0} + c = c$ and by definition of c also $c_i = v_i \neq 0$. Due to continuity of $\dot{x}_i$ there is an open interval J_i around t_1 such that $\forall t \in J_i : \operatorname{sign}(\dot{x}_i(t)) = v_i$.

[5] or w_i due to $i \in \operatorname{comm}(v,w)$

Observation 2: For $i \in \text{diff}(v, w)$ exists an interval J_i with $t_1 \in J_i = (a, b)$ such that $\forall t \in J_0 : \text{sign}(\ddot{x}_i(t)) = w_i$ holds. Furthermore, it holds $\forall t \in (a, t_1) : \text{sign}(\dot{x}(t)) = v_i$ and $\forall t \in (t_1, b) : \text{sign}(\dot{x}(t)) = w_i$.
Proof: For $t \in J_0$ it holds $\ddot{x}(t) = A \cdot \dot{x}(t)$ and $\dot{x}(t_1) = c$. By construction of A and c it holds $\text{sign}(\ddot{x}_i(t_1)) = w_i$.[6] Due to continuity there is an interval J_i around t_1 such that $\forall t \in J_i : \text{sign}(\ddot{x}_i(t)) = w_i$. For the second part of the proof we can use Theorem 4.1 applied to the function $\dot{x}_i(\cdot)$ on the interval J_i. Consequently for $t \in J_i$ smaller than t_1 the sign of $\dot{x}_i(t)$ is v_i, while for $t \in J_i$ bigger than t_1 the sign of $\dot{x}_i(t)$ is w_i.

To finish the proof we now choose $t_0 < t_1 < t_2$ such that $t_0, t_2 \in \bigcap_{i=0,\ldots,n} J_i$. Using the above observations we obtain finally

$$\forall t \in [t_0, t_1) : \text{sign}(\dot{x}(t)) = v,$$
$$\forall t \in (t_1, t_2] : \text{sign}(\dot{x}(t)) = w.$$

The function x is also obviously reasonable on the interval $\bigcap_{i=0,\ldots,n} J_i$. □

The above lemma shows that if for a tuple (v, w) in $\{-1, 1\}^n$ the condition in Proposition 4.1 is satisfied, then indeed (v, w) is a transition in $\tilde{G}_{\text{QSTG}}(\Sigma)$. It remains to show the other direction. That is, if (v, w) is a transition in $\tilde{G}_{\text{QSTG}}(\Sigma)$ then the condition in Proposition 4.1 is satisfied.

Lemma 4.3. Assume $(v, w) \in \tilde{E}_{\text{QSTG}}(\Sigma)$, $v \neq w$ then Condition (4.1.1) is satisfied.

Proof. We prove that the existence of a transition $(v, w) \in \tilde{E}_{\text{QSTG}}(\Sigma)$ implies Condition (4.1.1).
Due to Definition 4.5 of the graph $\tilde{G}_{\text{QSTG}}(\Sigma)$, there is only a transition (v, w) if Condition (4.1.2) is satisfied. Due to continuity of $\dot{x}$ we can conclude from Condition (4.1.2) that it holds for $i \in \text{diff}(v, w)$

$$\text{sign}(\dot{x}_i(t_1)) = 0. \tag{4.1.3}$$

Furthermore, since f in Condition (4.1.2) is a function in $\mathscr{M}(\Sigma)$, by definition $f \in C^1(X, \mathbb{R}^n)$. Therefore, also the second derivative of x exists.
Now consider an $i \in \text{diff}(v, w)$.
Case $v_i = 1$: Since $i \in \text{diff}(v, w)$ it holds $w_i = -1$. The function x_i obtains therefore a local maximum at t_1 (due to Theorem 4.1). Since $\dot{x}$ is continuously differentiable, it also holds $\text{sign}(\ddot{x}_i(t_1)) = w_i$. The case $\text{sign}(\ddot{x}_i(t_1)) = 0$ is excluded here, since x is a reasonable function what implies according to our definition that the second derivative cannot vanish in a local maximum.
Case $v_i = -1$: Since $i \in \text{diff}(v, w)$ it holds $w_i = 1$ and x_i obtains therefore a

[6] In detail: It holds $\ddot{x}(t_1) = A \cdot \dot{x}(t_1) = A \cdot c$ and therefore it holds also

$$\begin{aligned} \ddot{x}_i(t_1) &= \sum_{j \in [n]} a_{i,j} \cdot c_j \\ &= \underbrace{a_{i,k(i)} \cdot c_{k(i)}}_{= n \cdot \sigma_{i,k(i)} \cdot v_{k(i)} \in \{-n, n\}} + \underbrace{\sum_{j \in [n] \setminus \{k(i)\}} \overbrace{a_{i,j} c_j}^{\in \{-1,0,1\}}}_{\in [-(n-1),(n-1)]}, \end{aligned}$$

from which we can conclude $\text{sign}(\ddot{x}_i(t_1)) = \sigma_{i,k(i)} \cdot v_{k(i)}$. And by the definition of k in Lemma 4.1 it holds $w_i \cdot v_{k(i)} = \sigma_{i,k(i)}$ or equivalently $w_i = \sigma_{i,k(i)} \cdot v_{k(i)}$ (since $v_{k(i)} \cdot v_{k(i)} = 1$).

local minimum at t_1. With the same argumentation it holds $\operatorname{sign}([\ddot{x}_i(t_1)]) = 1 = w_i$.

To conclude the proof we notice that we also have due to the chain rule the equality $\ddot{x}(t_1) = J_f(x(t_1)) \cdot \dot{x}(t_1)$. I.e. for the components we have $\ddot{x}_i(t_1) = \nabla f_i(x(t_1)) \cdot \dot{x}(t_1)$. This implies:

$$w_i = \operatorname{sign}(\ddot{x}_i(t_1)) = \operatorname{sign}(\nabla f_i(x(t_1)) \cdot \dot{x}(t_1))$$

The above equality can only hold if $\exists j \in [n] : w_i = \operatorname{sign}(\partial_j f_i(x(t_1)) \cdot \dot{x}_j(t_1))$ is satisfied. Due to (4.1.3) the values of $\dot{x}_j(t_1) = c_j$ are zero for $j \in \operatorname{diff}(v, w)$. Therefore, it holds $j \in \operatorname{comm}(v, w)$. Then from $\operatorname{sign}(\partial_j f_i(x(t_1)) \cdot \dot{x}_j(t_1)) = \operatorname{sign}(\partial_j f_i(x(t_1))) \cdot \underbrace{\operatorname{sign}(\dot{x}_j(t_1))}_{v_j} = \sigma_{i,j} \cdot v_j$ the claim follows. □

Taking Lemma 4.2 and Lemma 4.3 together we proved now Proposition 4.1. The proof of Lemma 4.3 corresponds to the proof in [Eisenack, 2006, Prop. 2 (p. 25)]. However, there is the case where the second derivative vanishes is not excluded. Under these relaxed conditions the above lemma is wrong as the following example shows:

Example 4.4. Consider the monotonic ensemble $\mathscr{M}(\Sigma)$ with

$$\Sigma = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \end{pmatrix}.$$

According to Proposition 4.1 there should be no transition from $\begin{pmatrix} 1 \\ -1 \\ -1 \\ -1 \end{pmatrix}$ to $\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ in $\tilde{G}_{QSTG}(\Sigma)$ since $\sigma_{4,1} = 0$. However, under the relaxed conditions in [Eisenack, 2006, Prop. 2 (p. 25)] this is not true. To see this consider the ODE $\dot{x} = f(x)$ given by:

$$f(x) = \begin{pmatrix} x_1 \\ (x_1 - e)^2 x_1 + 10(x_1 - e) \\ 10(x_1 - e) \\ x_2 - x_3 \end{pmatrix}$$

The Jacobian matrix of f is given by

$$J_f = \begin{pmatrix} 1 & 0 & 0 & 0 \\ (x_1 - e)^2 + 2x_1(x_1 - e) + 10 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \end{pmatrix}.$$

It is easy to see that $\partial_1 f_2$ is everywhere positive.[7] Consequently, we have

$$\operatorname{sign}(J_f) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \end{pmatrix}$$

and therefore $f \in \mathscr{M}(\Sigma)$.

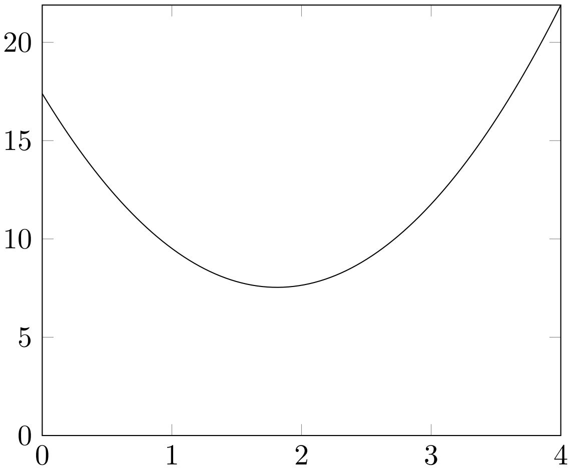


FIGURE 4.1.5. Plot of $(\partial_1 f_2)(x_1)$ in Example 4.4 on the interval $[0,4]$.

A solution is given by

$$x = \begin{pmatrix} e^t \\ \frac{1}{3}(e^t - e)^3 + 10(e^t - et) \\ 10(e^t - et) \\ \frac{1}{18}\left[2(e^{3t} - 3e^3 t) - 9e^{2t+1}\right] + e^{t+2} \end{pmatrix}$$

$$\Rightarrow \dot{x} = \begin{pmatrix} e^t \\ (e^t - e)^2 e^t + 10(e^t - e) \\ 10(e^t - e) \\ \frac{1}{3}(e^t - e)^3 \end{pmatrix}$$

$$\Rightarrow \ddot{x} = \begin{pmatrix} e^t \\ (e^t - e)^2 e^t + 2(e^t - e)e^{2t} + 10e^t \\ 10e^t \\ (e^t - e)^2 e^t \end{pmatrix}.$$

Now we notice that

$$\dot{x}(1) = \begin{pmatrix} e \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ and } \ddot{x}(1) = \begin{pmatrix} e \\ 10e \\ 10e \\ 0 \end{pmatrix}$$

holds. Therefore, we know that x_2 and x_3 take a local minimum in $t = 1$. Consequently, their derivatives change sign in $t = 1$ from negative to positive.

[7] Let $\varphi(t) = (t-e)^2 + 2t(t-e) + 10$. The first and second derivatives are given by $\dot{\varphi}(t) = 6t - 4e$ and $\ddot{\varphi}(t) = 6$. Since φ is quadratic, φ has a global minimum at $t_{\min} = \frac{4e}{6}$. Since $\varphi(t_{\min}) \approx 7,54$ φ is positive.

We also see that $\dot{x}_3(t) = \frac{1}{3}(e^t - e)^3$ has a sign change in $t = 1$ from negative to positive. Therefore, the derivative of the solution x has a sign change from $\begin{pmatrix} 1 \\ -1 \\ -1 \\ -1 \end{pmatrix}$ to $\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$, which is not in agreement with [Eisenack, 2006, Prop. 2 (p. 25)].

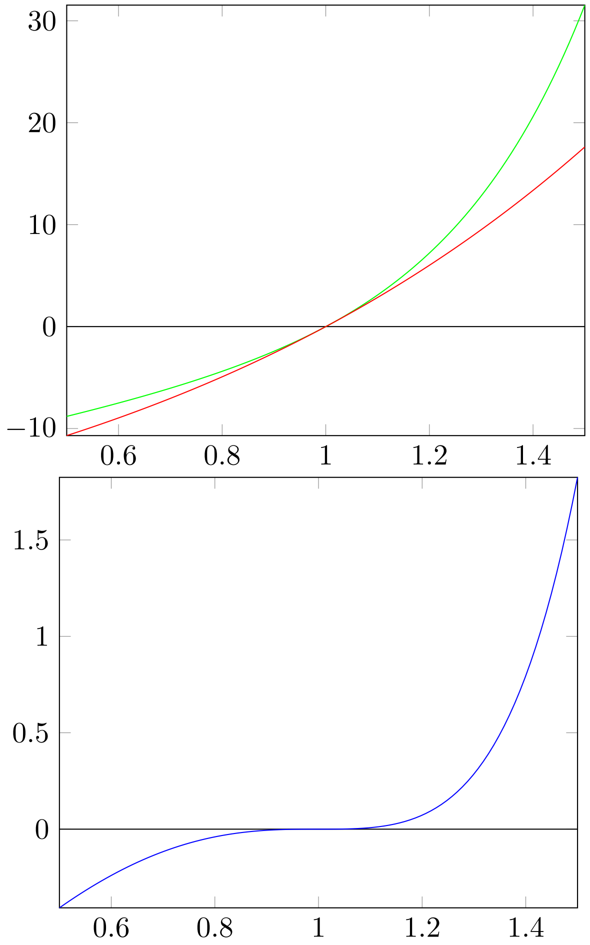


FIGURE 4.1.6. Plots of the derivatives of x_2 (green), x_3 (red) and x_4 (blue) in Example 4.4. All derivatives change sign at $t = 1$.

4.1.3. Boolean reformulation of the QSTG. The goal of this section is to show that we can consider the graph $\tilde{G}_{\text{QSTG}}(\Sigma)$ as a graph with set of nodes in $\{0,1\}^n$, if we replace the symbol -1 with 0. This reformulation will be useful in the rest of the chapter, when we compare the graph $\tilde{G}_{\text{QSTG}}(\Sigma)$ with a similar graph constructed for Boolean networks. We will show that the resulting graph is the following graph:

Definition 4.6. For a sign-matrix $\Sigma \in \{-1,0,1\}^{n\times n}$ we define the graph $G_{\text{QSTG}}(\Sigma) := \big(V_{\text{QSTG}}(\Sigma), E_{\text{QSTG}}(\Sigma)\big)$ with $V_{\text{QSTG}}(\Sigma) := \{0,1\}^n$. The edges of $G_{\text{QSTG}}(\Sigma)$ are given by the following condition, where we used the notation from Definition 1.15:

$$(v,w) \in E_{\text{QSTG}}(\Sigma) :\Leftrightarrow \bigwedge_{i \in \text{diff}(v,w)} \bigvee_{j \in \text{comm}(v,w)} v_i \oplus^{\sigma_{i,j}} v_j. \tag{4.1.4}$$

As usual, we call the edges of $G_{\text{QSTG}}(\Sigma)$ transitions and the nodes states.

As a first step we notice that we proved with Proposition 4.1 that the condition for a transition in the graph $\tilde{G}_{\text{QSTG}}(\Sigma)$ can be replaced by:

$$(v,w) \in \tilde{E}_{\text{QSTG}}(\Sigma) :\Leftrightarrow \bigwedge_{i \in \text{diff}(v,w)} \bigvee_{j \in \text{comm}(v,w)} (v_i \cdot v_j = -\sigma_{i,j}). \tag{4.1.5}$$

Note that we substituted w_i with $-v_i$ in (4.1.5) which is possible due to the equality $v_i = -w_i$ for $i \in \text{diff}(v,w)$. Note also that for all $v \in \tilde{V}_{\text{QSTG}}(\Sigma)$ it holds $(v,v) \in \tilde{E}_{\text{QSTG}}(\Sigma)$ since by convention the empty conjunction is always true (see Section 1.1.2).

Our goal is to show that the graphs $\tilde{G}_{\text{QSTG}}(\Sigma)$ and $G_{\text{QSTG}}(\Sigma)$ are isomorphic. Furthermore, the isomorphism is given by replacing the symbol -1 with 0. More formally, to show that the two graphs are isomorphic we introduce the mapping

$$\begin{aligned} \phi :& \{0,1\} \to \{-1,1\}, \\ x \mapsto & \begin{cases} -1 & \text{if } x = 0 \\ 1 & \text{if } x = 1 \end{cases}, \end{aligned}$$

and we define the function $\Phi : \{0,1\}^n \to \{-1,1\}^n$ with $\Phi : x \mapsto (\phi(x_1), \dots, \phi(x_n))$. This leads to the following proposition:

Proposition 4.2. The function Φ is a graph isomorphism (See Definition 1.11) from $G_{\text{QSTG}}(\Sigma)$ to $\tilde{G}_{\text{QSTG}}(\Sigma)$.

PROOF. It is easy to see that for any $a, b \in \{0,1\}$ and $\sigma \in \{-1,0,1\}$ the equivalence

$$a \oplus^{\sigma} b \Leftrightarrow (\phi(a) \cdot \phi(b) = -\sigma)$$

holds. To do so we need to prove the equivalence for the three cases $\sigma = -1$, $\sigma = 0$ and $\sigma = 1$ listed in the following table:

σ	$a \oplus^{\sigma} b$	$\phi(a) \cdot \phi(b) = -\sigma$
-1	$a \leftrightarrow b$	$\phi(a) \cdot \phi(b) = 1$
0	0	$\phi(a) \cdot \phi(b) = 0$
1	$a \oplus b$	$\phi(a) \cdot \phi(b) = -1$

From this observation it is easy to show that the graphs $\tilde{G}_{\text{QSTG}}(\Sigma)$ and $G_{\text{QSTG}}(\Sigma)$ are isomorphic. □

In the following we will identify the graphs $G_{\text{QSTG}}(\Sigma)$ and $\tilde{G}_{\text{QSTG}}(\Sigma)$ with each other and use for both the symbol $G_{\text{QSTG}}(\Sigma)$.

4.2. The qualitative state transition graph of a Boolean monotonic model pool

The goal of this section is to construct a graph $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ depending on a interaction graph represented as a sign matrix $\Sigma \in \{-1,0,1\}^{n \times n}$. The graph

$G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ will play a similar role as the graph $G_{\mathrm{QSTG}}(\Sigma)$ in Section 4.1. We will consider the set of Boolean functions agreeing with the interaction graph induced by Σ. We will refer to this set as *Boolean monotonic model pool* $\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$. Then we will abstract the dynamics – more specifically the ASTGs induced by the Boolean functions in $\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ – using quotient graphs. The quotient graphs we use for abstracting the dynamics are based on the images of f (see Section 3.2.2).

In Section 4.2.1 we introduce the graph $G^{\mathbb{B}}_{\mathrm{QSTG}}$ and state our main result. In Section 4.2.2 we give the proof of this result. The results here are a refinement of previous statements about representing a Boolean monotonic model pool as a state transition graph Schwieger and Siebert [2017, 2018].

4.2.1. A graph representation of a Boolean monotonic model pool. The object of our study is a set of Boolean functions $\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ agreeing with a sign matrix $\Sigma = (\sigma_{i,j})_{i,j\in[n]} \in \{-1,0,1\}^{n\times n}$. With "agree" we mean that for any Boolean function $f : \{0,1\}^n \to \{0,1\}^n$ in $\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ the global interaction graph $IG^{\mathrm{global}}(f)$ is a subgraph of the graph induced by the sign matrix Σ.[8] We write for this $IG^{\mathrm{global}}(f) \subseteq \Sigma$. More precisely, we define our set of Boolean functions $\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ in the following way:

Definition 4.7. For any $n \times n$ sign matrix $\Sigma = (\sigma_{i,j})_{i,j\in[n]}$, $\sigma_{i,j} \in \{-1,0,1\}$ we call the set

$$\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma) := \{f : \{0,1\}^n \to \{0,1\}^n | IG^{\mathrm{global}}(f) \subseteq \Sigma\}$$

the *Boolean monotonic model pool.* A function $f : \{0,1\}^n \to \{0,1\}^n$ for which a sign matrix Σ exists such that $f \in \overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ holds, is called *monotonic function.*

The notion of a monotonic function should not be confused with the notion of monotone functions as defined for example in Melliti et al. [2016], since the latter one is more restrictive. We illustrate the notions and results in this section with a running example:

Example 4.5. Consider the Boolean functions f and g defined as follows:

$$f : (x_1,x_2,x_3,x_4) \mapsto (x_2, x_1, x_4 \wedge (\neg x_2), \neg x_3)$$
$$g : (x_1,x_2,x_3,x_4) \mapsto (x_2, x_1, \neg x_2, \neg x_3)$$

It is easy to see that their (global) interaction graphs are given by

$$\Sigma = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

and

$$\tilde{\Sigma} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix}.$$

[8] I.e. let $G = ([n], E)$ be the graph such that Σ is its adjacency matrix. Then $IG^{\mathrm{global}}(f) \subseteq \Sigma :\Leftrightarrow E(IG^{\mathrm{global}}(f)) \subseteq E$.

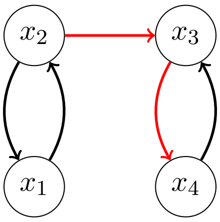
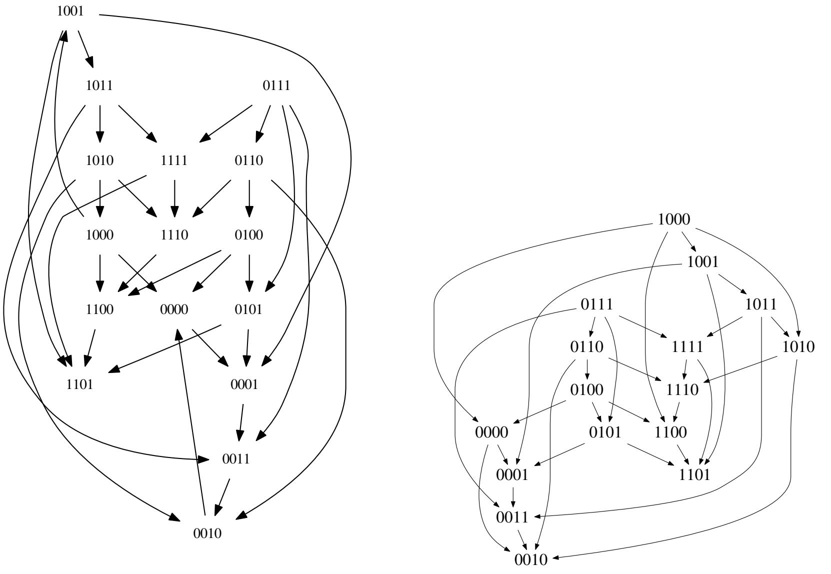


FIGURE 4.2.1. Interaction graph of the running example.



(A) $G_{\mathrm{async}}(f)$.

(B) $G_{\mathrm{async}}(g)$.

FIGURE 4.2.2. ASTGs in Example 4.5.

The graph induced by Σ is depicted in Figure 4.2.1. Both functions f and g are in the Boolean monotonic model pool $\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$. Their ASTGs are depicted in Figure 4.2.2a and Figure 4.2.2b. Clearly, the two BNs have different dynamics. For example $G_{\mathrm{async}}(f)$ has only the steady state 1101, while $G_{\mathrm{async}}(g)$ has additionally the steady state 0010. It is therefore not straightforward to see what characteristics of their dynamics are imposed by Σ.

To analyze the model pool $\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ we introduce a graph $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$. The meaning and relation of this graph to the model pool $\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ will be clarified later in Theorem 4.2.

Definition 4.8. For a sign matrix $\Sigma = (\sigma_{i,j})_{i,j\in[n]} \in \{-1,0,1\}^{n\times n}$ we define a graph $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma) = \big(V^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma), E^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)\big)$ with:

$$V^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma) = \{0,1\}^n$$

and

$$(v,w) \in E^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma) \quad :\Leftrightarrow \bigvee_{j\in[n]} \bigwedge_{i\in \mathrm{diff}(v,w)} \big(v_i \oplus^{\sigma_{i,j}} v_j\big), \tag{4.2.1}$$

where $a \oplus^1 b$, $a \oplus^0 b$ and $a \oplus^{-1} b$ for $a, b \in \{0,1\}$ were defined according to Definition 1.15.

Proposition 4.3. If Σ has no negative diagonal elements then (4.2.1) is equivalent to $\bigvee_{j\in \mathrm{comm}(v,w)} \bigwedge_{i\in \mathrm{diff}(v,w)} \big(v_i \oplus^{\sigma_{i,j}} v_j\big)$. This means it holds

$$(v,w) \in E^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma) \quad \Leftrightarrow \bigvee_{j\in \mathrm{comm}(v,w)} \bigwedge_{i\in \mathrm{diff}(v,w)} \big(v_i \oplus^{\sigma_{i,j}} v_j\big). \tag{4.2.2}$$

Proof. Since for any $i \in [n]$ it holds $\sigma_{i,i} \geq 0$ the logical expressions $v_i \oplus^{\sigma_{i,i}} v_i$ are always false. Consequently, we can restrict the set $[n]$ in the disjunction of (4.2.1) to $\mathrm{comm}(v,w)$. □

From the above definition we can derive immediately the following observations:

- If Condition (4.2.1) is satisfied there is a $j \in [n]$ such that for all $i \in \mathrm{diff}(v,w)$ the entries $\sigma_{i,j}$ are not zero.
- For each $v \in \{0,1\}^n$ there is a transition $(v,v) \in E^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$, since the set $\mathrm{diff}(v,v)$ will be empty in this case and Condition (4.2.1) is trivially satisfied for any index in $[n] = \mathrm{comm}(v,v)$.

To investigate the possible dynamics of models in $\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$, we are going to relate now the graph $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ with the model pool $\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$. To do so we use the equivalence relation based on the update function introduced in Section 3.2.2. For an update function $f \in \mathbb{B}(n,n)$ we defined the equivalence relation $\sim_f$ in such a way that for two states s, t in $\{0,1\}^n$ it holds

$$s \sim_f t :\Leftrightarrow f(s) = f(t).$$

We are interested in the quotient graphs $G_{\mathrm{async}}(f)/f := G_{\mathrm{async}}(f)/\sim_f$ induced by such equivalences $\sim_f$. In Section 3.2.2 we gave some intuition and simple properties for these types of quotient graphs.

We state now the main result of this section which explains the relation of the graph $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ and $\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$.

Theorem 4.2. Let $\Sigma \in \{-1,0,1\}^{n\times n}$ be a sign-matrix. The graph $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ is the union of all quotient graphs $G_{\mathrm{async}}(f)/f$ of Boolean functions in the model pool $\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$. I.e.

$$G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma) = \bigcup_{f\in\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)} G_{\mathrm{async}}(f)/f.$$

While the graph $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ can be easily constructed of the matrix Σ, the right hand side $\bigcup_{f\in\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)} G_{\mathrm{async}}(f)/f$ seems on the first glance very inaccessible. However, Theorem 4.2 states that it is in fact the same object as our graph $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$. Since $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ is constructed using only the

matrix Σ, this links the dynamical behavior of the models in $\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ to Σ. However, this relation between structure and dynamics takes the detour over the quotient graphs with respect to the update function.
We demonstrate the construction of the graph $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ with our running example:

Example 4.6. The graph $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ can be easily constructed from the interaction graph. In the case of our running example with $\Sigma = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix}$ the graph $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ is depicted in Figure 4.2.3. Assume we want to check if $\overbrace{1101}^{v} \to \overbrace{1111}^{w}$ is a transition of this graph: Since $w = v^{\{3\}}$ it holds $\operatorname{diff}(v,w) = \{3\}$, and therefore we obtain:

$$\bigvee_{j\in\{1,2,4\}} (v_3 \oplus^{\sigma_{3,j}} v_j).$$

This expression is equivalent to $(v_3 \oplus^1 v_4)$, which is obviously true. Therefore, $1101 \to 1111$ is a transition in $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ (see Figure 4.2.3).

In Figure 4.2.4 the quotient graphs $G_{\text{async}}(f)/f$ and $G_{\text{async}}(g)/g$ from Example 4.5 are depicted. We see that they are indeed subgraphs of $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$.

The graph $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ captures restrictions on the structure of the quotient graphs in the monotonic model pool. With the remarks about the relation of the ASTGs and their quotient graphs we can now summarize some simple consequences from Theorem 4.2:

Corollary 4.1. 1) If for two states $v, w \in \{0,1\}^n$ there is no path from v to w in $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$, then for no quotient graph $G_{\text{async}}(f)/f$ of Boolean functions in the model pool $\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ there is a path from v to w.
2) If $v \in \{0,1\}^n$ has no outgoing transitions in $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$, then for each quotient graph $G_{\text{async}}(f)/f$ in the model pool $\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ the state v can only occur as a steady state. Furthermore, either the set $f^{-1}(v)$ is empty or v is a steady state of $G_{\text{async}}(f)$.
3) Each trap set T in $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ corresponds to (a possibly empty) trap set in the quotient graphs $G_{\text{async}}(f)/f$ of the monotonic model pool. Consequently, also the preimage $f^{-1}(T)$ corresponds to a (a possibly empty) trap set in $G_{\text{async}}(f)$ for any $f \in \overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$.

4.2.2. Proof of the main result. We start by showing the inclusion

$$\bigcup_{f\in\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)} G_{\text{async}}(f)/f \subseteq G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$$

and continue afterwards with the opposite inclusion. The idea of the proof is to find a necessary condition for a transition in $G_{\text{async}}(f)/f$ for any $f \in \overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$, which we can express solely with the images of f representing the equivalence classes in $G_{\text{async}}(f)/f$. This condition will correspond exactly to Condition (4.2.1).

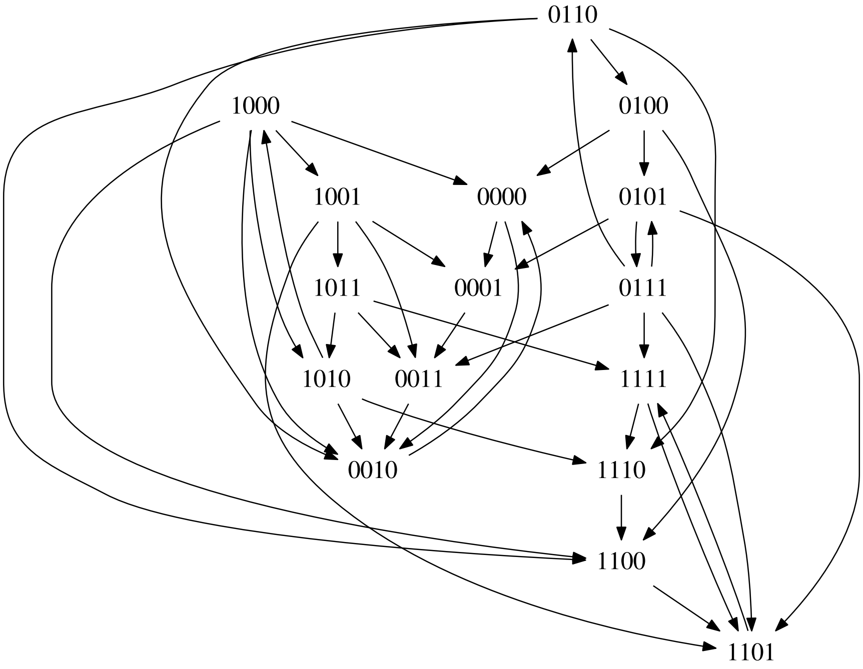
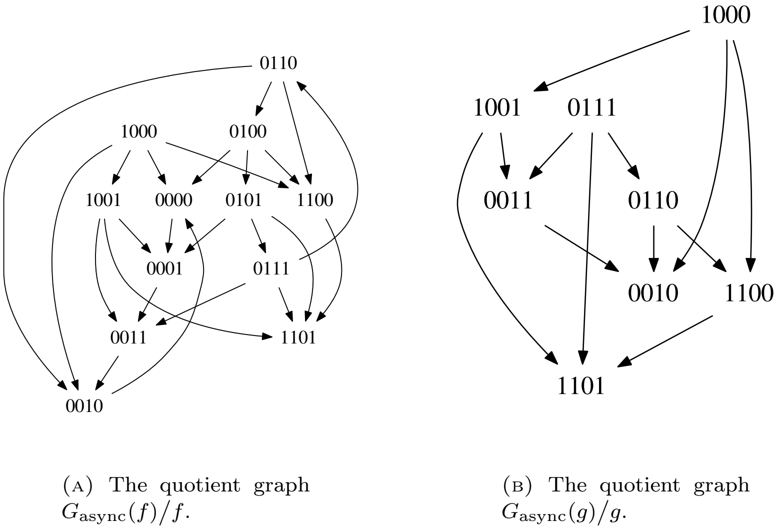


FIGURE 4.2.3. The graph $G_{\mathrm{QSTG}}^{\mathbb{B}}(\Sigma)$ from the running example (Example 4.6). Loops are omitted.



(A) The quotient graph $G_{\mathrm{async}}(f)/f$.

(B) The quotient graph $G_{\mathrm{async}}(g)/g$.

FIGURE 4.2.4. The quotient graphs of the functions defined in Example 4.5.

Since the graph $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ has for every state $v \in G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ a loop (v, v) and therefore the inclusion $\bigcup_{f \in \overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)} G_{\text{async}}(f)/f \subseteq G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ for these types of transitions is trivially true, it will be enough to focus for the first inclusion on transitions (v, w), $v, w \in \{0,1\}^n$ with $v \neq w$.

To ease later calculations we state the following lemma here, which can be easily checked by calculating the corresponding truth tables.

Lemma 4.4. For $\sigma \in \{-1, 1\}$ it holds $a \oplus^{\sigma} b \Leftrightarrow \frac{a-(a\oplus 1)}{(b\oplus 1)-b} = \frac{(a\oplus 1)-a}{b-(b\oplus 1)} = \sigma$.

Next, we prove a necessary condition for a transition in the quotient graph $G_{\text{async}}(f)/f$ with $f \in \overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$.

Lemma 4.5. Let $\Sigma = (\sigma_{i,j})_{i,j\in[n]} \in \{-1,0,1\}^{n\times n}$ be a $n\times n$ sign matrix. Assume for some $f \in \overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$, there are $s, t \in \{0,1\}^n$ s.t. $(s,t) \in E_{\text{async}}(f)$, $s \neq t$ with $f(s) = v$ and $f(t) = w$ (i.e. $(v, w) \in E(G_{\text{async}}(f)/f)$) then the Boolean expression

$$\bigvee_{j\in \text{diff}(s,t)} \bigwedge_{i \in \text{diff}(v,w)} (v_i \oplus^{\sigma_{i,j}} v_j) \tag{4.2.3}$$

is satisfied.

Proof. Due to $(s,t) \in E_{\text{async}}(f)$, $s \neq t$ it holds

$$\exists j \in \text{diff}(s,t) : s^{\{j\}} = t. \tag{4.2.4}$$

Let $j \in \text{diff}(s,t)$ such that Condition (4.2.4) is satisfied. Then for any $i \in \text{diff}(v,w)$ due to the definition of the interaction graph and the discrete derivatives $\partial_j f_i$ (see (1.3.1)) it holds first $\partial_j f_i(s) = \frac{f_i(t)-f_i(s)}{t_j - s_j} \neq 0$ and therefore $\sigma_{i,j} \neq 0$. Second, it holds:

$$\begin{aligned}
&\forall i \in \text{diff}(v,w) : \sigma_{i,j} = \frac{f_i(t) - f_i(s)}{t_j - s_j} = \frac{(f_i(s)\oplus 1) - f_i(s)}{t_j - (t_j \oplus 1)} \\
\Leftrightarrow &\forall i \in \text{diff}(v,w) : f_i(s) \oplus^{\sigma_{i,j}} t_j \\
\Leftrightarrow &\forall i \in \text{diff}(v,w) : f_i(s) \oplus^{\sigma_{i,j}} f_j(s) \\
\Leftrightarrow &\forall i \in \text{diff}(v,w) : v_i \oplus^{\sigma_{i,j}} v_j \\
\Leftrightarrow &\bigwedge_{i\in\text{diff}(v,w)} (v_i \oplus^{\sigma_{i,j}} v_j).
\end{aligned}$$

For the first equivalence we used Lemma 4.4 (with $a = f_i(s)$ and $b = t_j$). □

For any $f \in \overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ Lemma 4.5 gives us Condition (4.2.3) for the existence of a transition $(v, w) \in E(G_{\text{async}}(f)/f)$. However, Condition (4.2.3) depends not only on the states of $G_{\text{async}}(f)/f$ but also on the states and transitions in $G_{\text{async}}(f)$. In order to embed the graphs $G_{\text{async}}(f)/f$, $f \in \overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ into a super graph as in Theorem 4.2, we would like to obtain a condition similar to Condition (4.2.3) which only depends on the states in $G_{\text{async}}(f)/f$. We can do so by introducing the condition:

$$\bigvee_{j\in[n]} \bigwedge_{i\in\text{diff}(v,w)} (v_i \oplus^{\sigma_{i,j}} v_j). \tag{4.2.5}$$

We obtain the the following statement:

Proposition 4.4. Let Σ be a sign matrix. Assume there is $f \in \overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ such that $(v, w) \in E\big(G_{\text{async}}(f)/f\big)$ then

$$\bigvee_{j\in[n]} \bigwedge_{i\in\text{diff}(v,w)} (v_i \oplus^{\sigma_{i,j}} v_j). \tag{4.2.6}$$

Proof. Assume $(v, w) \in E\big(G_{\text{async}}(f)/f\big)$. Then there are two states s, t in $\{0, 1\}^n$ with $f(s) = v$ and $f(t) = w$. If $v = w$ holds, Condition (4.2.6) is trivially satisfied due to $\text{diff}(v, w) = \emptyset$. Otherwise, it holds $s \neq t$ and according to Lemma 4.5 the following condition is satisfied:

$$\bigvee_{j\in\text{diff}(s,t)} \bigwedge_{i\in\text{diff}(v,w)} (v_i \oplus^{\sigma_{i,j}} v_j).$$

Due to $\text{diff}(s, t) \subseteq [n]$ we obtain:

$$\bigvee_{j\in[n]} \bigwedge_{i\in\text{diff}(v,w)} (v_i \oplus^{\sigma_{i,j}} v_j).$$

□

As an immediate consequence of Proposition 4.4 the first inclusion in Theorem 4.2 follows:

Corollary 4.2. It holds

$$\bigcup_{f\in\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)} G_{\text{async}}(f)/f \subseteq G^{\mathbb{B}}_{\text{QSTG}}(\Sigma).$$

Proof. Set $\bigcup_{f\in\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)} G_{\text{async}}(f)/f := (V, E)$. We can identify the states of the quotient graphs $G_{\text{async}}(f)/f$ with its images under f. These images are in $\{0, 1\}^n$. Therefore, clearly $V \subseteq \{0, 1\}^n$ holds. For each transition $e = (v, w) \in G_{\text{async}}(f)/f$ according to the previous Proposition 4.4, Condition (4.2.1) in the definition of the graph $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ is satisfied. Therefore, the inclusion holds as well on the set of transitions. □

It remains to show $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma) \subseteq \bigcup_{f\in\overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)} G_{\text{async}}(f)/f$. For this purpose we construct for every transition $(v, w) \in E^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ a Boolean function $f \in \overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ such that its quotient graph $G_{\text{async}}(f)/f$ contains the transition (v, w).

Proposition 4.5. Assume $(v, v^A) \in E^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ with $A \subseteq [n]$. Then there is a Boolean function $f \in \overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ such that $(v, v^A) \in E(G_{\text{async}}(f)/f)$ holds.

Proof. Due to $(v, v^A) \in E^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ by the definition of $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ it holds:

$$\bigvee_{j\in[n]} \bigwedge_{i\in\text{diff}(v,v^A)} \big(v_i \oplus^{\sigma_{i,j}} v_j\big). \tag{4.2.7}$$

And due to Lemma 4.4 and $\text{diff}(v, v^A) = A$ the following is true:

$$\exists j \in [n] \forall i \in A : \big[(\sigma_{i,j} \neq 0) \wedge \big(\frac{v_i - (v_i \oplus 1)}{(v_j \oplus 1) - v_j} = \frac{(v_i \oplus 1) - v_i}{v_j - (v_j \oplus 1)} = \sigma_{i,j}\big)\big]. \tag{4.2.8}$$

We pick $j \in [n]$ such that $\forall i \in A : \big[(\sigma_{i,j} \neq 0) \wedge \big(\frac{v_i-(v_i\oplus 1)}{(v_j\oplus 1)-v_j} = \frac{(v_i\oplus 1)-v_i}{v_j-(v_j\oplus 1)} = \sigma_{i,j}\big)\big]$ is satisfied. Furthermore, we define for any $t \in \{0,1\}^n$ the function $f \in \mathbb{B}(n,n)$ in the following way[9]:

$$f(t) = \begin{cases} v & \text{if } t_j = v_j^{\{j\}} \\ v^A & \text{if } t_j = v_j \end{cases} \tag{4.2.9}$$

On the hyperplane defined by $\{t \in \{0,1\}^n | t_j = v_j^{\{j\}}\}$ every value is mapped to v by f, while on the hyperplane $\{t \in \{0,1\}^n | t_j = v_j\}$ every value is mapped to v^A. Clearly it holds $f(v^{\{j\}}) = v$. Therefore, there is a transition $(v^{\{j\}}, v) \in E_{\text{async}}(f)$. Since $f(v^{\{j\}}) = v$ and $f(v) = v^A$ holds, (v, v^A) is a transition in the quotient graph $G_{\text{async}}(f)/f$.
We claim now $IG^{\text{global}}(f) \subseteq \Sigma$ which is by definition equivalent to $f \in \overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$. The function f only changes along the j-axis. Therefore, we only need to check the partial derivatives $\partial_j f_i$ for $i \in [n]$. Furthermore, for any $i \in B$ the Boolean function f_i equals everywhere v_i. Therefore, we only need to check whether the partial derivatives $\partial_j f_i$ agree with Σ for $i \in A$.
Assume there is $s \in \{0,1\}^n$ such that $\partial_j f_i(s) \neq 0$ for some $i \in A$. Then due to the definition of the discrete derivative it holds:

$$0 \neq \frac{f_i(s^{\{j\}}) - f_i(s)}{s_j^{\{j\}} - s_j} = \partial_j f_i(s)$$

<u>**1. Case** $s_j = v_j^{\{j\}}$</u>: Then $f_i(s) = v_i$, $f_i(s^{\{j\}}) = v_i^A = 1 \oplus v_i$, $s_j = v_j^{\{j\}} = 1 \oplus v_j$ and $s_j^{\{j\}} = v_j$. This leads to:

$$\partial_j f_i(s) = \frac{f_i(s^{\{j\}}) - f_i(s)}{s_j^{\{j\}} - s_j} = \frac{1 \oplus v_i - v_i}{v_j - 1 \oplus v_j}$$

<u>**2. Case** $s_j = v_j$</u>: Then $f_i(s) = v_i^A = 1 \oplus v_i$, $f_i(s^{\{j\}}) = v_i$, $s_j = v_j$ and $s_j^{\{j\}} = v_j^{\{j\}} = 1 \oplus v_j$. This leads to:

$$\partial_j f_i(s) = \frac{f_i(s^{\{j\}}) - f_i(s)}{s_j^{\{j\}} - s_j} = \frac{v_i - v_i \oplus 1}{1 \oplus v_j - v_j}$$

And due to (4.2.8) in both cases it holds $\partial_j f_i(s) = \sigma_{i,j}$. □

The following example illustrates the above proof.

Example 4.7. Consider the transition $\overbrace{1101}^{=v} \to \overbrace{1111}^{w}$ in $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ of our running example. We want to construct a function f whose quotient graph $G_{\text{async}}(f)/f$ contains this transition. In the above theorem it holds $A = \{3\}$ and we can pick $j = 4$ as in Example 4.6. Consequently, we define

$$f(s) = \begin{cases} (1,1,0,1)^t & \text{if } s_4 = 0 \\ (1,1,1,1)^t & \text{if } s_4 = 1 \end{cases}.$$

[9] Note that we did not exclude the case $A = \emptyset$. Therefore, the function f could be constant, too.

I.e.

$$f(s) = \begin{pmatrix} 1 \\ 1 \\ s_4 \\ 1 \end{pmatrix}.$$

Due to Proposition 4.5 for every transition $(v, w) \in E^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ we can construct a Boolean function $f \in \overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ such that $G_{\text{async}}(f)/f$ contains the transition (v, w) and the states v, w. Therefore, we obtain the following corollary.

Corollary 4.3. It holds $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma) \subseteq \bigcup_{f \in \overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)} G_{\text{async}}(f)/f$.

Corollary 4.2 and Corollary 4.3 prove the equality in Theorem 4.2.

4.2.3. Some further remarks: other Boolean model pools. In this section we discuss in how far there is potential to refine the results in Section 4.2. In Section 4.2 we saw that we can relate the dynamics of different models in a Boolean monotonic model pool to a state transition graph $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$. This relation was elucidated by Theorem 4.2. This result raises the question if it would be possible to restrict the model pool we consider here to any subset $M \subset \overline{\mathscr{M}}_{\mathbb{B}}(\Sigma)$ and prove a result similar to Theorem 4.2. A natural candidate would be

$$\mathscr{M}_{\mathbb{B}}(\Sigma) := \left\{ f : \{0,1\}^n \to \{0,1\}^n | IG^{\text{global}}(f) = \Sigma \right\}.$$

I.e. we would need to prove that

$$G^{\mathbb{B}}_{\text{QSTG}}(\Sigma) = \bigcup_{f \in \mathscr{M}_{\mathbb{B}}(\Sigma)} G_{\text{async}}(f)/f \tag{4.2.10}$$

holds. However, the following example shows that we can find a sign matrix Σ for which

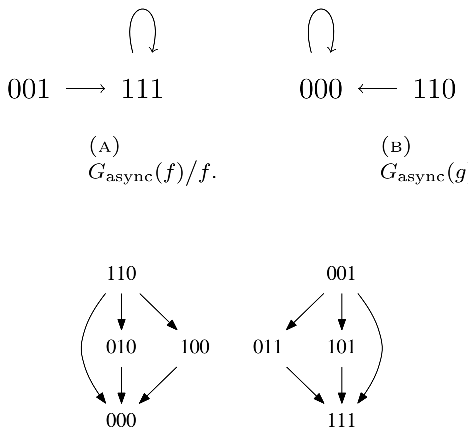
$$\bigcup_{f \in \mathscr{M}_{\mathbb{B}}(\Sigma)} G_{\text{async}}(f)/f \neq G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$$

holds. The reason for this is roughly speaking that while the update scheme is asynchronous enforcing transitions of the form$(v, v^{\{i\}})$ the change in the image of f on such transitions i.e. $(f(v), f(v^{\{i\}})$ is still synchronous i.e. the change in one components affects all values in f simultaneously.

We can use this idea to construct an example in which (4.2.10) does not hold. We choose a very simple matrix Σ, where a change let us say in the third variable enforces a change in the first and second one (I.e. 001 → 111 in $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$). In the graph $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ that we choose such a transition would imply also transitions of the form 001 → 011 and 001 → 101 (see also Proposition 4.8 in Section 4.3). However, as we will see the latter transitions can not occur in the quotient graphs in the Boolean monotonic model pool.

Example 4.8. Consider the matrix

$$\Sigma = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$



(A) $G_{\text{async}}(f)/f$.

(B) $G_{\text{async}}(g)/g$.

(C) The graph $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ without self-loops.

FIGURE 4.2.5. Illustration of Example 4.8.

In this specific case it is easy to describe the model pool $\mathscr{M}_{\mathbb{B}}(\Sigma)$.
Claim: In the model pool $\mathscr{M}_{\mathbb{B}}(\Sigma)$ consists only of two functions, namely

$$f(s) = \begin{pmatrix} s_3 \\ s_3 \\ 1 \end{pmatrix},$$

$$g(s) = \begin{pmatrix} s_3 \\ s_3 \\ 0 \end{pmatrix}.$$

Proof: Due to the structure of the global interaction graph for any function $h \in \mathscr{M}_{\mathbb{B}}(\Sigma)$ the third component h_3 must be constant and the first and second component must be a function of s_3 - i.e. $h_1(s_3), h_2(s_3)$. There are only two non-constant functions $\{0,1\} \to \{0,1\}$ and only one agrees with the sign structure of Σ.
It is now easy to see that there is a transition $(v,w) \in E^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ which occurs in none of the two quotient graphs $G_{\text{async}}(f)/f$ and $G_{\text{async}}(g)/g$: In $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ is the transitions $001 \to 101$, since

$$\overbrace{v_1}^{=0} \oplus \overbrace{\sigma_{1,3}}^{=1} \overbrace{v_3}^{=1}$$

is satisfied, which is the condition in Definition 4.8 for this transition. But $001 \to 101$ is neither a transition in $G_{\text{async}}(f)/f$ nor in $G_{\text{async}}(g)/g$. The follows from the fact that 101 is not in the image of f and g.

4.2.4. $G^{\mathbb{B}}_{\mathbf{QSTG}}(\Sigma)$ is a subgraph of $G_{\mathbf{QSTG}}(\Sigma)$. The condition for a transition in $G_{\text{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ are very similar. Let us consider the case where $\Sigma \in \{-1,0,1\}^{n\times n}$ is sign matrix with non-negative diagonal elements. In this case the condition for a transition in $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ is given by (4.2.2) in Proposition 4.3. After switching the order of the conjunction and disjunction symbol in (4.2.2) one obtains the condition for a transition in $G_{\mathrm{QSTG}}(\Sigma)$:

$$\begin{aligned} &(v,w) \in E_{\mathrm{QSTG}}(\Sigma) \\ \Leftrightarrow &\bigwedge_{i \in \mathrm{diff}(v,w)} \bigvee_{j \in \mathrm{comm}(v,w)} \left(v_i \oplus^{\sigma_{i,j}} v_j\right) \end{aligned} \tag{4.2.11}$$

Therefore, we have:

Corollary 4.4. For a $n \times n$ sign matrix with non-negative diagonal elements it holds $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma) \subseteq G_{\mathrm{QSTG}}(\Sigma)$.

Proof. Condition (4.2.1) implies Condition (4.2.11). Therefore, the inclusion holds since every transition of $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ is a transition in $G_{\mathrm{QSTG}}(\Sigma)$. □

This shows that the graphs $G_{\mathrm{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ are in fact not that different from each other. We will learn more about this in the next section.

4.3. The skeleton of the qualitative state transition graphs

The goal of this section is to show that we can reduce the transitions in $G_{\mathrm{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ to transitions between states with Hamming distance one from each other without changing reachability properties of the graphs. This reduction is useful for several reasons. Firstly, we are mostly interested in structures of the state space which are defined in terms of reachability properties (e.g. trap sets, steady states). Secondly, we can argue that transitions where only one component changes at a time are much more likely to observe, since the different components of the modeled regulatory network are likely to interact with each other on different timescales Thomas [1991]. Lastly, we can represent these transitions as an ASTG. We will call this ASTG the skeleton of Σ. Interestingly, we will see that for a sign-matrix Σ with non-negative diagonal the skeleton of $G_{\mathrm{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ is identical. This implies especially that a state w is reachable from a state v in $G_{\mathrm{QSTG}}(\Sigma)$ if and only if the same holds true in $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$. Analogous to the construction of $G_{\mathrm{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ the skeleton, which will be denoted by $G_{\mathrm{async}}(f^{\Sigma})$, can be constructed directly from Σ. This allows us to analyze the skeleton – an asynchronous BN with ASTG $G_{\mathrm{async}}(f^{\Sigma})$ – instead of the graphs $G_{\mathrm{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$. Another aspect is that we can reconstruct the sign matrix Σ from the skeleton. This is useful if the sign matrix Σ is unknown and we wish to reconstruct it from observations. The results in this section are partially published in Schwieger and Siebert [2018]. We start now with the construction of the graph $G_{\mathrm{async}}(f^{\Sigma})$.

In Section 4.1 and Section 4.2 we defined the graphs $G_{\mathrm{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ in such a way that for a transition $e = (v,w)$, $v, w \in \{0,1\}^n$ with

$\mathrm{diff}(v, w) = A$ the following conditions hold:

$$(v, w) \in E_{\mathrm{QSTG}}(\Sigma) \Leftrightarrow \bigwedge_{i \in \mathrm{diff}(v,w)} \bigvee_{j \in \mathrm{comm}(v,w)} v_i \oplus^{\sigma_{i,j}} v_j,$$

$$(v, w) \in E^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma) \Leftrightarrow \bigvee_{j \in [n]} \bigwedge_{i \in \mathrm{diff}(v,w)} (v_i \oplus^{\sigma_{i,j}} v_j).$$

This implies that there are potentially transitions (v, w) in $G_{\mathrm{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ where the set $\mathrm{diff}(v, w)$ has cardinality bigger than one. However, we want to represent the graphs $G_{\mathrm{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ as an ASTG of a Boolean function. Therefore, we show now in the sequel that we can delete the transitions (v, w) with $|\mathrm{diff}(v, w)| > 1$ without loosing information about reachability.[10]

For the transitions (v, w) for which the set $\mathrm{diff}(v, w)$ has cardinality one, i.e. $\mathrm{diff}(v, w) = \{i\}$, the conditions for a transition in $G_{\mathrm{QSTG}}(\Sigma)$ where Σ has no negative diagonal elements and in $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ for any $\Sigma \in \{-1, 0, 1\}^{n \times n}$ are given by the following logical formula:

$$\bigvee_{j \in [n]} (v_i \oplus^{\sigma_{i,j}} v_j). \tag{4.3.1}$$

We can then use (4.3.1) to construct the Boolean update function of the corresponding ASTG. We explain this now in detail.

4.3.1. The skeleton of the graph $G_{\mathrm{QSTG}}(\Sigma)$. Since the condition for a transition in the graph $G_{\mathrm{QSTG}}(\Sigma)$ is independent of the diagonal elements of Σ we can without loss of generality assume that for the graph $G_{\mathrm{QSTG}}(\Sigma)$ the diagonal elements of Σ are zero. The following proposition shows that we can restrict ourselves to transitions with Hamming distance one in $G_{\mathrm{QSTG}}(\Sigma)$ without loosing information about reachability:

Proposition 4.6. Assume there is a transition $(v, v^A) \in E_{\mathrm{QSTG}}(\Sigma)$ with $A \subseteq [n]$. Then for each $B \subseteq A$ it holds

$$(v, v^B) \in E_{\mathrm{QSTG}}(\Sigma),$$
$$(v^B, v^A) \in E_{\mathrm{QSTG}}(\Sigma).$$

Proof. We show first $(v, v^B) \in E_{\mathrm{QSTG}}(\Sigma)$: Since $B \subseteq A \Leftrightarrow [n]\backslash A \subseteq [n]\backslash B$ the following implications hold true:

$$\begin{aligned}&(v, v^A) \in E_{\mathrm{QSTG}}(\Sigma)\\ \Rightarrow &\bigwedge_{i \in A} \bigvee_{j \in [n]\backslash A} (v_i \oplus^{\sigma_{i,j}} v_j)\\ \Rightarrow &\bigwedge_{i \in B} \bigvee_{j \in [n]\backslash B} (v_i \oplus^{\sigma_{i,j}} v_j)\\ \Rightarrow &(v, v^B) \in E_{\mathrm{QSTG}}(\Sigma)\end{aligned}$$

[10] $|\cdot|$ denotes the cardinality of a set. (See also Section 1.1)

For the second part we know due to $(v, v^A) \in E_{\text{QSTG}}(\Sigma)$:

$$\bigwedge_{i \in A} \bigvee_{j \in [n] \setminus A} \left(v_i \oplus^{\sigma_{i,j}} v_j\right).$$

Let us call $C := \text{diff}(v^B, v^A) = A \setminus B$. We need to show

$$\bigwedge_{i \in C} \bigvee_{j \in [n] \setminus C} \left(v_i^B \oplus^{\sigma_{i,j}} v_j^B\right).$$

Before we start we make two observations:
<u>1. Observation</u>: $\forall j \in [n] \setminus A : v_j = v_j^A = v_j^B$
<u>2. Observation</u>: $\forall i \in C : v_i = v_i^B$

Now we are ready to prove the statement of the Proposition. Since $B \cup C = A$ we have $C \subseteq A$. It follows

$$\begin{aligned} &\bigwedge_{i \in A} \bigvee_{j \in [n] \setminus A} \left(v_i \oplus^{\sigma_{i,j}} v_j\right) \\ \Rightarrow &\bigwedge_{i \in C} \bigvee_{j \in [n] \setminus A} \left(v_i \oplus^{\sigma_{i,j}} v_j\right). \end{aligned}$$

And due to Observation 1 and 2 we obtain:

$$\bigwedge_{i \in C} \bigvee_{j \in [n] \setminus A} \left(v_i^B \oplus^{\sigma_{i,j}} v_j^B\right).$$

Since $C \subseteq A \Leftrightarrow [n] \setminus A \subseteq [n] \setminus C$ finally we showed:

$$\bigwedge_{i \in C} \bigvee_{j \in [n] \setminus C} \left(v_i^B \oplus^{\sigma_{i,j}} v_j^B\right) \Leftrightarrow (v^B, v^A) \in E_{\text{QSTG}}(\Sigma).$$

□

Now for transitions with Hamming distance one the condition for a transition $(v, v^{\{k\}})$ with $k \in [n]$ in $G_{\text{QSTG}}(\Sigma)$ takes the form

$$\begin{aligned} (v, v^{\{k\}}) \in E_{\text{QSTG}}(\Sigma) &\Leftrightarrow \bigwedge_{i \in \text{diff}(v,w)} \bigvee_{j \in \text{comm}(v,w)} v_i \oplus^{\sigma_{i,j}} v_j \\ &\Leftrightarrow \bigvee_{j \in [n] \setminus \{k\}} v_k \oplus^{\sigma_{k,j}} v_j \\ &\Leftrightarrow \bigvee_{j \in [n]} v_k \oplus^{\sigma_{k,j}} v_j, \end{aligned}$$

where the last equivalence holds since $v_k \oplus^{\sigma_{k,k}} v_k$ is always false.

4.3.2. The skeleton of the graph $G^{\mathbb{B}}_{\mathbf{QSTG}}(\Sigma)$. It can easily be shown that Proposition 4.6 holds as well for the graph $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ if Σ contains no negative diagonal elements. However, in the general case we need to be careful as the following example shows.

Example 4.9. Consider the sign matrix $\Sigma = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix}$. The graph $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ depicted in Figure 4.3.1 contains the transition $10 \to 01$[11] and $10 \to 00$. However, it does not contain the transition $00 \to 01$[12]. This shows that the analogous statement of Proposition 4.6 is not correct for the graph $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ if Σ contains negative diagonal elements.

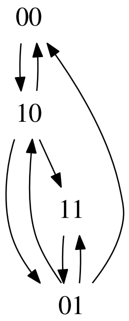


FIGURE 4.3.1. The graph $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ in Example 4.9.

However, we can prove a similar result. We start with the following lemma which gives us a simplified version of the condition for a transition in $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ if no component with a negative self-loop changes along a transition in $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$:

Lemma 4.6. Let $\Sigma \in \{-1,0,1\}^n$ be a sign matrix. Assume there is a transition $(v, v^A) \in E^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ with $A \subseteq [n]$, $A \neq \emptyset$ and for all $i \in A$ it holds $\sigma_{i,i} \geq 0$. Then it holds

$$(v, v^A) \in E^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma) \Leftrightarrow \bigvee_{j \in [n]\setminus A} \bigwedge_{i \in A} \left(v_i \oplus^{\sigma_{i,j}} v_j\right). \tag{4.3.2}$$

PROOF. It holds

$$(v, v^A) \in E^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma) \Leftrightarrow \bigvee_{j \in [n]} \bigwedge_{i \in A} \left(v_i \oplus^{\sigma_{i,j}} v_j\right).$$

If $j \in A$ then the expression $\left(v_j \oplus^{\sigma_{j,j}} v_j\right)$ is always false. This shows the equivalence:

$$\bigvee_{j \in [n]} \bigwedge_{i \in A} \left(v_i \oplus^{\sigma_{i,j}} v_j\right) \Leftrightarrow \bigvee_{j \in [n]\setminus A} \bigwedge_{i \in A} \left(v_i \oplus^{\sigma_{i,j}} v_j\right)$$

□

[11] For $v = 10$ and $w = 01$ the condition $\bigvee_{j \in \{1,2\}} \bigwedge_{i \in \{1,2\}} \left(v_i \oplus^{\sigma_{i,j}} v_j\right)$ is equivalent to $(v_1 \oplus^{\sigma_{1,1}} v_1) \wedge (v_2 \oplus^{\sigma_{2,1}} v_1) = (v_1 \oplus^{-1} v_1) \wedge (v_2 \oplus^{1} v_1)$, which is true.

[12] For $v = 00$ and $w = 01$ the condition $\bigvee_{j \in \{1,2\}} \bigwedge_{i \in \{2\}} \left(v_i \oplus^{\sigma_{i,j}} v_j\right)$ is equivalent to $\underbrace{(v_2 \oplus^{\sigma_{2,1}} v_1)}_{=0} \vee \underbrace{(v_2 \oplus^{\sigma_{2,2}} v_2)}_{=0}$, which is not satisfied.

Furthermore, for arbitrary transitions in $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ we have the following lemma:

Lemma 4.7. Assume there is a transition $(v, v^A) \in E^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ with $A \subseteq [n]$. Then for any subset $B \subseteq A$ there is a transition $(v, v^B) \in E^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$.

Proof. Since (v, v^A) is a transition in $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ it holds $\bigvee_{j\in[n]} \bigwedge_{i\in A} (v_i \oplus^{\sigma_{i,j}} v_j)$. It follows that $\bigvee_{j\in[n]} \bigwedge_{i\in B} \left(v_i \oplus^{\sigma_{i,j}} v_j\right)$ is satisfied for any $B \subseteq A$, which is equivalent to $(v, v^B) \in E^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$. □

Using Lemma 4.6 and Lemma 4.7 we can prove now the analogous result to Proposition 4.6 for transitions not involving negatively autoregulated components:

Proposition 4.7. Let $\Sigma \in \{-1, 0, 1\}^n$ be a sign matrix. Assume there is a transition $(v, v^A) \in E^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ with $A \subseteq [n]$, $A \neq \emptyset$ and for all $i \in A$ it holds $\sigma_{i,i} \geq 0$. Then for each $B \subseteq A$, $A \neq B \neq \emptyset$:

$$(v, v^B) \in E^{\mathbb{B}}_{\text{QSTG}}(\Sigma),$$
$$(v^B, v^A) \in E^{\mathbb{B}}_{\text{QSTG}}(\Sigma).$$

Proof. This is a consequence of Lemma 4.6 and Lemma 4.7. The proof is analogous to Proposition 4.6. □

Next, we make the observation that for a component $k \in [n]$ with a negative self-loop (i.e. $\sigma_{k,k} = -1$) each state v in $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ has a transition where the k-th component changes:

Lemma 4.8. For any $v \in \{0, 1\}^n$, $k \in [n]$ with $\sigma_{k,k} = -1$ it holds

$$(v, v^{\{k\}}) \in E^{\mathbb{B}}_{\text{QSTG}}(\Sigma).$$

Proof. This is true since in this case $v_k \oplus^{\sigma_{k,k}} v_k$ is always satisfied and furthermore we have the implications:

$$\begin{aligned} & v_k \oplus^{\sigma_{k,k}} v_k \\ \Rightarrow & \bigvee_{j\in[n]} \bigwedge_{i\in\{k\}} \left(v_i \oplus^{\sigma_{i,j}} v_j\right) \\ \Leftrightarrow & (v, v^{\{k\}}) \in E^{\mathbb{B}}_{\text{QSTG}}(\Sigma) \end{aligned}$$

□

Proposition 4.8. For any transition (v, w) in $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ with $v \neq w$ there is a path from v to w in $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ consisting of transitions of Hamming distance one.

Proof. Let $w = v^A$ with $\emptyset \neq A \subset [n]$. We define $A = B \cup C$ with $B = \{i \in A | \sigma_{i,i} \geq 0\}$ and $C = A \backslash B$. First we notice that due to Lemma 4.7 there is a transition (v, v^B) in $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$. Due to Proposition 4.7 we know that there is path consisting of transitions with Hamming distance one from v to v^B in $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$. Next, by applying Lemma 4.8 iteratively we obtain the same result for v^B and $v^A = (v^B)^C$. This proves the claim. □

Proposition 4.8 and Proposition 4.6 show that we can restrict ourselves to the transitions induced by the sets $\{1\}, \dots, \{n\}$ without loosing information about reachability of states in $G_{\text{QSTG}}(\Sigma)$ or $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$. We call the resulting graph $G(\Sigma) = (V(\Sigma), E(\Sigma))$ the *skeleton of* Σ, where $V(\Sigma) := \{0,1\}^n$ and for $v \in \{0,1\}^n, i \in [n]$ it holds

$$(v, v^{\{i\}}) \in E(\Sigma) :\Leftrightarrow \bigvee_{j\in[n]} v_i \oplus^{\sigma_{i,j}} v_j. \tag{4.3.3}$$

Since each transition in $G(\Sigma)$ is of the form $(v, v^{\{i\}})$, it is possible to represent this graph as the ASTG of a suitable Boolean function, which we will call f^{Σ}.

Remark 4.3. Note that the graph $G(\Sigma)$ has in contrast to the graphs $G_{\text{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ no self-loops. However, this is only a technical detail since this self-loops can easily be restored from $G(\Sigma)$ if necessary.

Indeed, due to Lemma 3.2 we can define f^{Σ} in the following way:

Definition 4.9. We define the function $f^{\Sigma} : \{0,1\}^n \to \{0,1\}^n$ as

$$f^{\Sigma} = \text{cond}^{g^{\Sigma}}, \tag{4.3.4}$$

where $g_i^{\Sigma}(v) := \bigvee_{j\in[n]} v_i \oplus^{\sigma_{i,j}} v_j$.

Due to Lemma 3.2 it holds $G(\Sigma) = G_{\text{async}}(f^{\Sigma})$. Therefore, we will denote the skeleton of Σ from now on with $G_{\text{async}}(f^{\Sigma})$. For two states $v, w \in \{0,1\}^n$, $v \neq w$ there is a directed path in $G_{\text{async}}(f^{\Sigma})$ if and only if there is a directed path in $G_{\text{QSTG}}(\Sigma)$. The same holds true for $G_{\text{async}}(f^{\Sigma})$ and $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$. The reduction of $G_{\text{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ to $G_{\text{async}}(f^{\Sigma})$ has not only the advantage that the graph $G_{\text{async}}(f^{\Sigma})$ can have significantly less transitions, but also that certain structural features of $G_{\text{async}}(f^{\Sigma})$ can be deduced directly from f^{Σ}. This includes trap sets (see Definition 1.7), attractors (see Definition 1.9), trap spaces (see Definition 1.27) and no-return sets (see Definition 1.8).

We illustrate the construction of f^{Σ} with the following example:

Example 4.10. Let us construct the functions g^{Σ} and f^{Σ} in Definition 4.9 from the sign matrix Σ in Example 4.1. We obtain

$$\begin{aligned} g_1^{\Sigma}(v) &= \bigvee_{j\in\{4\}} \left(v_1 \oplus^{\sigma_{1,j}} v_j\right) \\ &= \neg(v_1 \oplus v_4), \\[1em] g_2^{\Sigma}(v) &= \bigvee_{j\in\{1\}} \left(v_2 \oplus^{\sigma_{2,j}} v_j\right) \\ &= v_1 \oplus v_2, \\[1em] g_3^{\Sigma}(v) &= \bigvee_{j\in\{2,4\}} \left(v_3 \oplus^{\sigma_{3,j}} v_j\right) \\ &= (v_2 \oplus v_3) \vee \neg(v_3 \oplus v_4), \end{aligned}$$

$$\begin{aligned} g_4^\Sigma(v) &= \bigvee_{j\in\{3\}} \left(v_4 \oplus^{\sigma_{4,j}} v_j\right) \\ &= \neg(v_3 \oplus v_4) \end{aligned}$$

and

$$\begin{aligned} f_1^\Sigma(v) &= \neg v_4, \\ f_2^\Sigma(v) &= v_1, \\ f_3^\Sigma(v) &= v_3 \oplus \big((v_2 \oplus v_3) \vee \neg(v_3 \oplus v_4)\big), \\ f_4^\Sigma(v) &= \neg v_3. \end{aligned}$$

In Figure 4.3.2 the skeleton is depicted. The graph $G_{\mathrm{QSTG}}(\Sigma)$, depicted in Figure 4.1.3, has 50 transitions and its skeleton $G_{\mathrm{async}}(f^\Sigma)$ has 36 transitions.

We summarize the results in the following proposition:

Corollary 4.5 (of Proposition 4.8 and Proposition 4.6)**.** Let Σ be a sign matrix in $\{-1,0,1\}^{n\times n}$. Then we have for $v, w \in \{0,1\}^n$ the equivalences:

$$\begin{aligned} & v \rightsquigarrow w \text{ in } G_{\mathrm{async}}(f^\Sigma) \\ \Leftrightarrow & v \rightsquigarrow w \text{ in } G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma). \end{aligned} \tag{4.3.5}$$

If Σ has no negative diagonal elements additionally it holds.

$$\begin{aligned} & v \rightsquigarrow w \text{ in } G_{\mathrm{QSTG}}(\Sigma) \\ \Leftrightarrow & v \rightsquigarrow w \text{ in } G_{\mathrm{async}}(f^\Sigma) \\ \Leftrightarrow & v \rightsquigarrow w \text{ in } G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma). \end{aligned} \tag{4.3.6}$$

This implies for $T \subseteq \{0,1\}^n$ and $* \in \{$trap set, trap space, steady state, attractor, weak/strong basin of attraction of a trap set, no-return set$\}$ also:

$$\begin{aligned} & T \text{ is a } * \text{ in } G_{\mathrm{QSTG}}(\Sigma) \\ \Leftrightarrow & T \text{ is a } * \text{ in } G_{\mathrm{async}}(f^\Sigma) \\ \Leftrightarrow & T \text{ is a } * \text{ in } G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma). \end{aligned}$$

Proof. The equivalences in (4.3.5) and (4.3.6) are an immediate consequence of Proposition 4.6 and Proposition 4.8. For each transition (v, v^A), $v \in \{0,1\}^n$, $A := \{i_1,\dots,i_k\} \subseteq [n]$, $k \in [n]$, $A \neq \emptyset$ in the graph $G_{\mathrm{QSTG}}(\Sigma)$ or $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ there exists a path from v to v^A of the form $v \to v^{\{i_1\}} \to v^{\{i_1,i_2\}} \to \dots \to v^A$. Consequently, it holds $\big(v \rightsquigarrow w \text{ in } G_{\mathrm{QSTG}}(\Sigma)\big) \Leftrightarrow \big(v \rightsquigarrow w \text{ in } G_{\mathrm{async}}(f^\Sigma)\big)$ and $\big(v \rightsquigarrow w \text{ in } G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)\big) \Leftrightarrow \big(v \rightsquigarrow w \text{ in } G_{\mathrm{async}}(f^\Sigma)\big)$. If Σ has no negative diagonal elements the graphs $G_{\mathrm{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ have the same skeleton. This shows the equivalence (4.3.6). The second part of the claim follows immediately from this. □

Corollary 4.5 shows that we can shift the analysis of the graphs $G_{\mathrm{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\mathrm{QSTG}}(\Sigma)$ to the skeleton $G_{\mathrm{async}}(f^\Sigma)$. For many relevant structures such as trap spaces, attractors, steady states there exist specialized algorithms for BNs Klarner et al. [2015], Garg et al. [2008], Hopfensitz et al. [2013], Klarner and Siebert [2015]. Furthermore, network reduction techniques such as Veliz-Cuba et al. [2014a], Naldi et al. [2011], Zañudo and Albert [2013] can be used

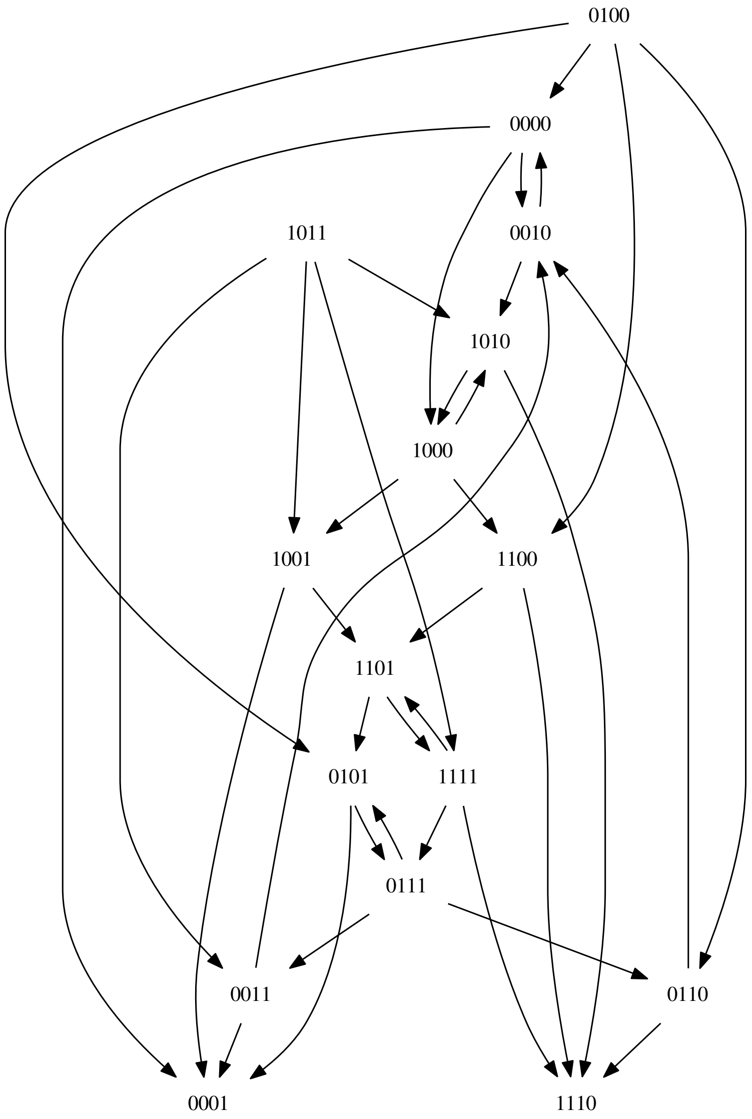


FIGURE 4.3.2. Skeleton in Example 4.10.

to simplify the model. Corollary 4.5 allows to use all these algorithms and ideas to analyze the graphs $G_{\text{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$. We also refer to Schwieger and Siebert [2018] for further ideas such as consistency with time series data. We will see in the rest of this thesis several applications of the graph $G_{\text{async}}(f^{\Sigma})$.

4.4. Reconstruction of the interaction graph

In this section we are going to demonstrate that the techniques developed in Section 4.3 can be used to infer interaction graphs. To do so we assume that the measured data corresponds to transitions in $G_{\text{QSTG}}(\Sigma)$ or more precisely to transitions in $G_{\text{async}}(f^{\Sigma})$. By using for example finite differences the slopes of the courses of concentrations of different species (e.g. proteins or mRNA) on different time intervals can be approximated. We assume that these approximated slopes correspond to the states of the graph $G_{\text{QSTG}}(\Sigma)$. In this way we obtain a set of transitions in $G_{\text{QSTG}}(\Sigma)$ for an unknown sign matrix Σ which we want to reconstruct. Using Proposition 4.6 we can assume further that all these transitions stem from the graph $G_{\text{async}}(f^{\Sigma})$. We will show in Proposition 4.9 that this approach allows us in principle to reconstruct the non-diagonal elements of the sign matrix Σ if the amount of data is sufficient. If this is not the case several sign matrices can agree with the given data. We can make additional assumptions about the structure of Σ such as sparsity to restrict the amount of results in this case.

By construction it makes only sense to consider interaction graphs stemming from monotonic functions. This is not a severe restriction since most models of GRNs are assumed to be monotonic. Also due to the way the graph $G_{\text{QSTG}}(\Sigma)$ is constructed in Section 4.1 it is not possible to reconstruct the diagonal elements. Consequently, if possible models containing autoregulation should not be too simplified such that concentrations of intermediate products necessary for the autoregulation can be compared to the predictions of the model as well.

To formalize our approach we define for $a \in \{-1,0,1\}^n$ the set of sign matrices with diagonal a, i.e.

$$\mathbb{M}^a = \big\{M \in \{-1,0,1\}^{n\times n} | \operatorname{diag}(M) = a\big\}$$

We want to show that it is in principle possible to reconstruct the sign matrix Σ from the skeleton $G_{\text{async}}(f^{\Sigma})$ or from the graph $G_{\text{QSTG}}(\Sigma)$. The idea is to prove that the map which attributes to a sign matrix Σ its skeleton f^{Σ}, i.e.

$$\begin{aligned} \Phi : \mathbb{M}^a &\to \mathbb{B}(n,n), \\ \Sigma &\mapsto f^{\Sigma}, \end{aligned} \tag{4.4.1}$$

is injective. Since the correspondence between the update function of an ASTG and its ASTG is as well bijective, this means that we can use the transitions of $G_{\text{async}}(f^{\Sigma})$ to reconstruct Σ. However, it will be rarely the case that all transitions of $G_{\text{async}}(f^{\Sigma})$ can be observed. This is a general problem of network inference Laubenbacher and Stigler [2004], Vera-Licona et al. [2014], Liang et al. [1998], Barman and Kwon [2017], Imani and Braga-Neto [2015], Lähdesmäki et al. [2003]. Additional assumptions are made to overcome this problem. A frequent one is that the interaction graph of the regulatory network is sparse. That means models in agreement with the measured transitions with smaller interaction graphs are to be preferred.

We start by proving that the function Φ is injective. This shows that it is in principle possible to restore the interaction graph from its skeleton.

4.4.1. There is a one-to-one correspondence between Σ, $G_{\text{async}}(f^{\Sigma})$, $G_{\text{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$. Let $\Sigma = (\sigma_{i,j})_{i,j\in[n]} \in \{-1,0,1\}^n$ be a sign matrix with non-negative diagonal elements. To show that the map Φ defined in (4.4.1) is injective we first observe that the component functions Φ_i,

$i \in [n]$ depend only on the elements of Σ in the i-th row which are not on the diagonal. We therefore define a family of maps

$$\begin{aligned} &\phi^i : \{-1,0,1\}^n \to \mathbb{B}(n,1), \\ (4.4.2) \qquad &\begin{pmatrix}\sigma_1 & \dots & \sigma_n\end{pmatrix} \mapsto \Big(x \mapsto x_i \oplus \big[\bigvee_{j \in [n]\setminus\{i\}} (x_i \oplus^{\sigma_j} x_j)\big]\Big) \end{aligned}$$

and consider these maps on the hyperplanes where the i-th component is fixed. Let us denote with σ^i the i-th row vector of the matrix Σ. It holds according to Definition 4.9 that $\Phi_i(\Sigma) = (f^{\Sigma})_i = \phi^i(\sigma^i)$. Each function ϕ^i does not depend on the i-th entry of the sign vector σ^i. This allows us to deduce that Φ is injective on the set $\mathbb{M}^a$, $a \in \{-1,0,1\}^n$ if for all $i \in [n]$ the functions ϕ^i are injective on such hyperplanes.

To prove that ϕ^i, $i \in [n]$ is injective we define for an arbitrary subset W of the Cartesian product of n sets S a set $W^{i,s}$ consisting of the elements of W for which the i-th component is $s \in S$. More precisely, for any set S and $W \subseteq \prod_{i=1}^n S$ the set $W^{i,s}$, $s \in S$ is defined as follows

$$W^{i,s} := \big\{ w \in W | w_i = s \big\}.$$

Using this notation we can prove now that the maps ϕ^i are injective. The idea of the proof is to construct for any two distinct sign vectors σ^1, σ^2 with $\sigma_i^1 = \sigma_i^2$ a point $a = a(\sigma^1, \sigma^2) \in \{0,1\}^n$ for which $\phi^i(\sigma^1)(a) \neq \phi^i(\sigma^2)(a)$ holds.

Lemma 4.9. The function ϕ^i is injective for $i \in [n]$ on the hyperplanes $H^{i,-1}$, $H^{i,0}$, $H^{i,1}$ with $H = \{-1,0,1\}^n$.

PROOF. For a fixed $i \in [n]$ assume there are two vectors $\sigma^1, \sigma^2 \in H^{i,*}$ such that $\sigma^1 \neq \sigma^2$, $* \in \big\{-1,0,1\big\}$. Furthermore, assume w.l.o.g. that there is an index $j \in [n]\setminus\{i\}$ such that $\sigma_j^2 \neq 0$ and $\sigma_j^1 \neq \sigma_j^2$.[13] We want to show $f_i^1 := \phi^i(\sigma^1) \neq \phi^i(\sigma^2) =: f_i^2$. Let a be a vector to be chosen in $\{0,1\}^n$. For $j \in [n]\setminus\{i\}$ we define $C_j^1 := (a_i \oplus^{\sigma_j^1} a_j)$ and $C_j^2 := (a_i \oplus^{\sigma_j^2} a_j)$. Then we can consider the disjunctions $C^1 := \bigvee_{j \in [n]\setminus\{i\}} C_j^1$ and $C^2 := \bigvee_{j \in [n]\setminus\{i\}} C_j^2$. It is easy to see that we can choose a in such a way that for all $j \in [n]\setminus\{i\}$ the expression C_j^1 evaluates to zero, while for at least one $j \in \mathrm{diff}(\sigma^1, \sigma^2)$ the expression C_j^2 evaluates to one.[14] Then the disjunction C^1 is false and C^2 is true. Consequently, it holds $\phi^i(\sigma^1)(a) \neq \phi^i(\sigma^2)(a)$ and the claim follows. □

Using the fact that the i-th component functions of Φ depend only on the i-th row and that these maps are injective on the hyperplane where the i-th component is fixed we arrive at the following proposition.

[13] Otherwise, just change the roles of σ^1 and σ^2.

[14] For example choose $a_i = 1$ on $H^{i,1}$ and for $j \in [n]\setminus\{i\}$ define

$$a_j := \begin{cases} 0 & \sigma_j^1 = -1 \\ 1 & \sigma_j^1 = 1 \\ 0 & \sigma_j^1 = 0 \text{ and } \sigma_j^2 = 1 \\ 1 & \sigma_j^1 = 0 \text{ and } \sigma_j^2 \in \{-1,0\} \end{cases}.$$

Proposition 4.9. The map Φ is injective on the set of sign matrices $\mathbb{M}^a$ for any $a \in \{-1,0,1\}^n$.

PROOF. According to Lemma 4.9 the maps ϕ^i are all injective on $H^{i,a}$. Consequently Φ is injective. □

We remark also that from the fact that Φ is injective a few interesting consequences can be drawn.

Corollary 4.6. Let $\Sigma \in \{-1,0,1\}^{n\times n}$ be a sign matrix with non-negative diagonal elements. Then the correspondences between the graphs $G_{\text{async}}(f^\Sigma)$, $G_{\text{QSTG}}(\Sigma)$, $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ are all one-to-one.

PROOF. By deleting all transitions between states with Hamming distance strictly bigger than one we obtain a mapping from $G_{\text{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ to f^Σ. Therefore, we obtain the following diagram for f^Σ and $G_{\text{QSTG}}(\Sigma)$:

$$\begin{tikzcd}
f^\Sigma & G_{\text{QSTG}}(\Sigma) \arrow[l] \\
\Sigma \in \mathbb{M}^a \arrow[u, "\Phi"'] \arrow[ur] &
\end{tikzcd}$$

(similar for f^Σ and $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$). Since Φ is injective all the other mappings in the diagram need to be injective as well. Consequently, also the correspondence between $G_{\text{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ is one-to-one. □

4.4.2. A network inference algorithm. We show now how based on the result of the last section a simple network inference algorithm can be constructed. For this purpose we define some metrics to compare a graph with a second graph which represents our knowledge about the dynamics of a BN.

Definition 4.10 (see also Barman and Kwon [2017])**.** For a graph $\tilde{G} = (V, \tilde{E})$ we define the *recall* with respect to a graph $G = (V, E)$ as

$$\text{Recall} = \frac{|\tilde{E} \bigcap E|}{|E|}.$$

If $\tilde{G}$ and G are ASTGs we also define for $i \in [n]$

$$\text{Recall}_i = \frac{\left|\left\{(v, v^{\{i\}}) \in \tilde{E} \bigcap E | v \in \{0,1\}^n\right\}\right|}{\left|\left\{(v, v^{\{i\}}) \in E | v \in \{0,1\}^n\right\}\right|}.$$

In the second part of the above definition we consider only transitions where the i-th component changes. In both parts of the definition the maximal recall rate is one which is the case if $E \subseteq \tilde{E}$.

We can now put everything together and describe a simple algorithm that will take as an input some discretized time series data that we assume will have the form of an ASTG. The idea of our algorithm (Algorithm 1) is to iterate over all possible skeletons beginning with the ones induced by the sparsest sign vectors. If there are several solutions induced by equally "likely" sign vectors (i.e. sign vectors whose support has the same cardinality)

Algorithm 1 G stores the measured transitions, i is the component of the interaction graph whose incoming edges we want to interpolate, m is the maximal indegree of the network.

```
function INTERPOLATE $f_i^{\Sigma}$(Graph $G$ (an ASTG storing the given data), $i \in [n]$, $m \in [n-1]$)
    $maxr \leftarrow -1$
    $maxk \leftarrow 1$
    $results \leftarrow list()$
    for $k = 1..m$ do
        for $\sigma \in \{-1,0,1\}^n$, $\sigma_i = 1$ with $|\{j \in [n] | \sigma_j \neq 0\}| = k$ do
            $f_i \leftarrow \phi^i(\sigma_i)$
            $r \leftarrow \mathrm{recall}_i(f_i, G)$
            if $r = maxr$ and $k = maxk$ then
                $results.append(f_i)$
            end if
            if $r > maxr$ then
                $maxk \leftarrow k$
                $maxr \leftarrow r$
                $results \leftarrow list(\{f_i\})$
            end if
        end for
        if $maxr = 1$ then
            return $results$
        end if
    end for
    return $results$
end function
```

that agree equally well with the given data the algorithm will return all possible solutions. Each of the solutions of the algorithm corresponds due to Proposition 4.9 uniquely to a sign vector σ using the map $(\phi^i)^{-1}$, $i \in [n]$.[15]

4.4.3. Numerical experiments. We consider the Boolean function

$$f : (x_1, x_2, x_3, x_4) \mapsto (\neg x_4, \neg x_1, x_1 \wedge x_2, x_3).$$

Its interaction graph $IG^{\mathrm{global}}(f)$ is depicted in Figure 4.4.1. We use the transformation algorithm presented in Chapter 2 with randomly chosen Hill exponents between 1 and 5, threshold between 0.2 and 0.8, and lifetimes between 1 and 5. We simulate this ODE-system on the interval $[0,5]$ and pick 100 equidistant time points. We use then Algorithm 1 to reconstruct the interaction graph $IG^{\mathrm{global}}(f)$.

As a first step we use finite differences to obtain a sequence of sign vectors. A negative sign is replaced with zero, a positive with one. Subsequent equal vectors are merged. In this way we obtain the following path: 1101 → 1100 → 1000 → 1010 → 1011. Since we do not have more information about the behavior of the system, we choose this path as the ASTG representing the gold standard. Applying Algorithm 1 we obtain Table 4.4.1. Since our

[15] up to the distinguished coordinate σ_i

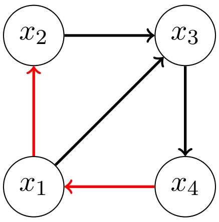


FIGURE 4.4.1. Interaction graph we wish to reconstruct. Red edges denote inhibitions.

component	results	corresponding edges
x_1	x_1	-
x_2	$x_2 \oplus (x_2 \oplus^{-1} x_1)$, $x_2 \oplus (x_2 \oplus^{1} x_3)$,$x_2 \oplus (x_2 \oplus^{1} x_4)$	$x_1 \rightarrow x_2$, $x_3 \rightarrow x_2$,$x_4 \rightarrow x_2$
x_3	$x_3 \oplus (x_3 \oplus^{1} x_1)$, $x_3 \oplus (x_3 \oplus^{-1} x_2)$, $x_3 \oplus (x_3 \oplus^{-1} x_4)$	$x_1 \rightarrow x_3$, $x_2 \rightarrow x_3$, $x_4 \rightarrow x_3$
x_4	$x_4 \oplus (x_4 \oplus^{-1} x_2)$, $x_4 \oplus (x_4 \oplus^{1} x_3)$	$x_2 \rightarrow x_4$, $x_3 \rightarrow x_4$

TABLE 4.4.1. Red edges denote inhibitory effects, black edges denote activating effects.

data contains no change in the first component it is reasonable to assume that it is not influenced by any other components. For the second component we obtain three alternative regulators. Either the first down-regulates, or the third or fourth component up-regulates the second component. While only the first one is correct, the other two options are also plausible since there exists a positive path from the third or fourth component to the first one in the interaction graph of the gold standard depicted in Figure 4.4.1. For the other components the situation is similar. Only the inhibitory edge from the second two the fourth component is not reflected in any negative path in $IG^{\text{global}}(f)$.

4.5. Discussion

In this chapter we investigated how complete sets of models of GRNs can be analyzed with a state transition graph derived directly from an interaction graph. We considered ODE models in Section 4.1 and asynchronous BNs in Section 4.2. The approach in the two cases is similar. By abstracting the dynamics of specific models it was possible to construct a finite state transition graph capturing restrictions on the dynamics of the models.

In Section 4.1 the abstraction of the solutions was the direction of the slopes of the components of the solution (i.e. increasing or decreasing). Similar in Section 4.2, Boolean models – represented by a function $f \in$

$\mathbb{B}(n,n)$ – were abstracted using quotient graphs $G_{\text{async}}(f)/f$. This allowed us to construct in Section 4.1 the graph $G_{\text{QSTG}}(\Sigma)$ and in Section 4.2 the graph $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$. The main result in both cases is that these graphs can be directly constructed from the sign matrix Σ without considering specific models.

In both cases, this makes it possible to deduce properties of the set of models from the graph $G_{\text{QSTG}}(\Sigma)$ or $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$. By construction, these properties are mostly negative, i.e. they concern the limits of possible behaviors of the models.

In Section 4.3 we unified the analysis of the graphs $G_{\text{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ by showing that they have a common skeleton. This skeleton is the ASTG of a Boolean function $f^{\Sigma} \in \mathbb{B}(n,n)$ obtained directly from the sign matrix Σ. The ASTG $G_{\text{async}}(f^{\Sigma})$ contains already all information about the reachability of the states in $G_{\text{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ to each other.

This allows us to apply theoretical results about BNs to the graphs $G_{\text{QSTG}}(\Sigma)$ and $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ or to analyze their structure with software tools developed for asynchronous BNs. For possible applications of our results we refer also to Corollary 4.1 and the list of simple statements about the relationship between the quotient graphs $G_{\text{async}}(f)/f$ and $G_{\text{async}}(f)$ in the end of Section 4.2.1. These results could in the future be exploited in the analysis of Boolean monotonic model pools when enumerating models explicitly is computationally too costly.

In Section 4.4 we investigated how the interaction graph can be reconstructed from the skeleton. We proved that this is possible by showing that the interaction graph and the skeleton are in a one-to-one correspondence if we neglect the elements on the diagonal. Afterwards we gave a naive algorithm which identifies the most sparse skeletons agreeing with given data.

There are a few aspects of the results in this chapter worth investigating in the future. The considered construction of the graphs $G_{\text{QSTG}}(\Sigma)$ or $G^{\mathbb{B}}_{\text{QSTG}}(\Sigma)$ can lead to graphs that are very well connected and therefore impose not many restrictions on the dynamics of the models. In some instances it is therefore not possible to draw many conclusions about the considered set of models. To overcome this problem it could be interesting to consider the informations about different time scales at which reactions are taking place. In many GRNs some reactions are very fast, while others are significantly slower (See e.g. [Alon, 2007, Table 2.1.]). Replacing Condition (4.1.4) by a more restrictive version, taking such information into account, could therefore be a possible path to improve the results presented here. We could work also directly on the skeleton and use results on different update schemes (see e.g. Noual [2012]) using priority classes. Also the idea of using the skeleton for network inference was just demonstrated here in a proof-of-concept style and deserves in my opinion further investigation. Another related topic that we are going to investigate in Chapter 5 is how we can use the representation of the interaction graph as a skeleton to develop techniques to distinguish different interaction graphs based on time series data or steady states. Interaction graphs are a static description of the interactions in the modeled regulatory network. However, such a static description is not always straightforward to verify especially if components are only indirectly measured. The skeleton constructed here we believe is a useful tool to overcome this bridge containing the relevant and most likely observed possible sign changes which are in agreement with the interaction graphs.

CHAPTER 5

Methods for applications

Contents

5.1. Distinguishing Boolean models with criteria based on states or transitions 102
5.1.1. Criteria based on states for measuring components 102
5.1.2. Criteria based on transitions for measuring components 108
5.1.3. Showcase: cell-fate decision 110
5.1.4. An algebraic reformulation of the State-Discrimination-Problem 114
5.2. Distinguishing Boolean models with criteria based on walks 121
5.2.1. Formalization 121
5.2.2. Application to four models of a Cytokinin signaling pathway 126
5.3. Discussion 130

In Chapter 4 we investigated how we can describe sets of ODEs and Boolean models that have a common structure. If we are not sure about the structure of a GRN – i.e. several interaction graphs are plausible – we could compare the corresponding skeletons introduced in the last chapter and see how well they agree with measured data. However, in most applications we do not have access to the state of the complete network. Therefore, comparing different ASTGs is not always straightforward. Normally, it is only possible to measure a small fraction of the components of the regulatory network. In this chapter we will present several new ideas and approaches to distinguish Boolean models based on a limited number of components and how to select components that could be measured to differentiate between BNs.

The topic of experiment design is not new and has been studied for example in Thiele et al. [2018] and many others. Here, we do not investigate any changes of the network by knock-out experiments. We focus solely on the question which components should be observed if we do not influence the behavior of the regulatory network. Also the design of Boolean classifiers in synthetic biology, considered for example in Mohammadi et al. [2017], Becker et al. [2018] is related. In these works Boolean classifiers with a specific structure are searched which can be implemented as synthetic biological circuits. Here, we do not make such structural assumptions on the Boolean functions representing the classifiers. We will formalize our criteria using projections of the states or transitions of a BN. Therefore, there is also a link to network reduction (see e.g. Paulevé and Richard [2012], Naldi et al. [2011]). Network reduction methods often try as well to conserve the number of steady states or more generally attractors, which is in principle a similar problem to the one considered in this chapter.

Depending on what aspects of a Boolean network we consider as predictive features of the model different criteria for selecting components might be reasonable. In Section 5.1 we introduce two criteria to distinguish Boolean models based on states and transitions. The first three subsections of Section 5.1 are based on a joint work with Claudine Chaouiya and Pedro T. Monteiro which took place during a research stay at the INESC-ID[1] in Lisbon in spring 2018. In Section 5.1.3 these methods will be applied to a model for cell-fate decision Calzone et al. [2010]. In Section 5.1.4 we explain how the content of Section 5.1.1 and Section 5.1.2 could be described in the language of algebraic geometry.

In Section 5.2 we give ideas how sequences of states can be used to distinguish models. In this section we also demonstrate the methods on four possible interaction graphs of a Cytokinin signaling pathway suggested in Topcu-Alici [2012]. To do so the suggested interaction graphs are translated into the skeletons introduced in the previous chapter.

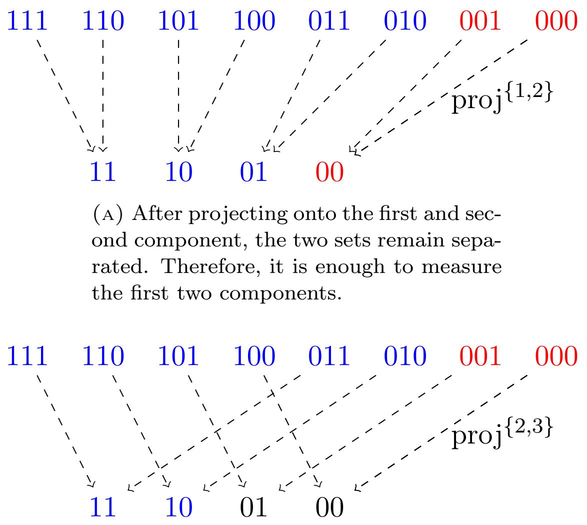
The ideas in this chapter are meant to be a primer for further research and should demonstrate how the results of this thesis can be exploited in the future in applications. We will give a few showcases. However, we will not go into detail about implementation and algorithmic aspects and leave this for future work. The methods here are not restricted to Boolean models arising from interaction graphs considered in Chapter 4. In principle, they are not even restricted to the scenario where different Boolean models are compared. Normally, asynchronous BNs, due to their nondeterminism, predict several different possible behaviors of the modeled network. Consequently, the methods in this chapter can be used to distinguish between different such possible behaviors of ASTGs as well. In Section 5.1.3 for example we will demonstrate how different groups of steady states can be distinguished based on a set of markers in a Boolean model for cell-fate decision.

5.1. Distinguishing Boolean models with criteria based on states or transitions

In this section we consider two criteria to select components we can measure to distinguish between models. The criteria will be based on projections of states or transitions of a BN. Afterwards, we outline algorithms to find these sets of components.

5.1.1. Criteria based on states for measuring components. In this section we consider the situation where we are given two disjunct sets of states A_1, A_2 in $\{0,1\}^n$. These sets could for example represent two sets of steady states of a BN we wish to distinguish (see the showcase in Section 5.1.3). Now assume we are given any state in the union of the two sets – i.e. $s \in A_1 \cup A_2$. We want to know what components we need to know such that we can decide for any state $s \in A_1 \cup A_2$ whether it belongs to A_1 or A_2. Let us look at an example to clarify what we mean with this:

[1] Instituto de Engenharia de Sistemas e Computadores, Investigação e Desenvolvimento em Lisboa



(A) After projecting onto the first and second component, the two sets remain separated. Therefore, it is enough to measure the first two components.

(B) After projecting these sets of states onto the second and third component, it is not possible anymore to distinguish the two sets from each other.

FIGURE 5.1.1. Illustration of Example 5.1. The state space $\{0,1\}^3$ is divided into the sets $\{000, 001\}$ (the red states) and $\{0,1\}^3 \backslash \{000, 001\}$ (the blue states).

Example 5.1. Consider the set of states depicted in Figure 5.1.1. The two groups of states, depicted in the figure, remain separated after the projection depicted in Figure 5.1.1a. Therefore, it is possible to decide their membership to the red or blue sets solely based on the first two components. In Figure 5.1.1b we see that measuring the second and third component is in this example not sufficient to decide the membership.

We formalize this in the following problem:

Problem 5.1. For two disjoint sets A_1, A_2 in $\{0,1\}^n$ find all sets of indices $\emptyset \neq I \subseteq [n]$ such that $\mathrm{proj}^I(A_1) \cap \mathrm{proj}^I(A_2) = \emptyset$ holds.

An alternative formulation, which is sometimes more convenient, is the following:

Problem 5.2 (State-Discrimination-Problem)**.** For a given set of states $\emptyset \neq A \subset B \subseteq \{0,1\}^n$ we search one or more sets $\emptyset \neq I \subseteq [n]$ such that for all $v \in B$ it holds

$$v \in A \Leftrightarrow \mathrm{proj}^I(v) \in \mathrm{proj}^I(A). \tag{5.1.1}$$

By setting $A = A_1$ and $B = A_1 \cup A_2$ or $A_1 = A$ and $A_2 = B \backslash A$ it is clear that these problems can be converted into each other without any problems. A trivial solution of the State-Discrimination-Problem is always $I = [n]$. However, in general we are interested in solutions which contain the least amount of components, i.e. a set $I \subseteq [n]$ with minimal cardinality. At this point we need to be careful, since it is in principle possible that different

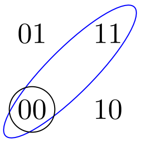
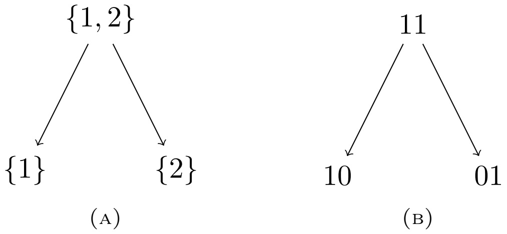


FIGURE 5.1.2. Illustration of Example 5.2: In order to identify $A = \{00\}$ in the set $B = \{00, 11\}$ we can either use the first or second component.



(A)

(B)

FIGURE 5.1.3. The set of solutions $\mathscr{A}(A, B)$ in Example 5.2 represented as Hasse diagram. On the left the elements of $\mathscr{A}(A, B)$ are sets, on the right they are represented as Boolean vectors. The minimal elements of $\mathscr{A}(A, B)$ are the Boolean vectors 10 and 01.

solutions have minimal cardinality. It is easy to construct such an example in $\{0, 1\}^2$.

Example 5.2. Let

$$A = \{00\},$$
$$B = \{00, 11\},$$

be two sets in $\{0, 1\}^2$. It is easy to see that $I = \{1\}$ and $I = \{2\}$ are both solutions with minimal cardinality to the State-Discrimination-Problem (see Figure 5.1.2).

We want now to investigate the solutions of the State-Discrimination-Problem. For $\emptyset \neq A \subset B \subseteq \{0, 1\}^n$ let us denote the solutions of the State-Discrimination-Problem with $\mathscr{A}(A, B)$. It is clear that if $I \in \mathscr{A}(A, B)$ is such a solution then for any index $j \in [n]$ also $I \cup \{j\} \in \mathscr{A}(A, B)$. Therefore, we can represent $\mathscr{A}(A, B)$ completely by its inclusion-minimal elements and equip it with the partial ordering given by the set inclusion. In order to find the solutions of the State-Discrimination-Problem we will give another reformulation of the above problem. The first step is to identify the elements of $\mathscr{A}(A, B)$ with Boolean vectors in $\{0, 1\}^n$ using the bijection

$$\begin{aligned} \mathscr{P}([n]) &\to \{0, 1\}^n, \\ I &\mapsto 0^I. \end{aligned} \tag{5.1.2}$$

Therefore, we can think of the set $\mathscr{A}(A, B)$ as a subset of $\{0, 1\}^n$. In Figure 5.1.3 we see the solutions $\mathscr{A}(A, B)$ in Example 5.2. On the right we see the set $\mathscr{A}(A, B)$ represented with Boolean vectors.

We construct now a Boolean function in $\mathbb{B}(n,1)$ that evaluates to one for each Boolean vector representing a solution of the State-Discrimination-Problem. We call such a function an indicator function, defined formally in the following way:

Definition 5.1. For a set $A \subseteq \{0,1\}^n$ we denote with ind^A the following function:

$$\begin{aligned} \mathrm{ind}^A : \{0,1\}^n &\to \{0,1\}, \\ s &\mapsto \begin{cases} 1 & \text{if } s \in A \\ 0 & \text{if } s \notin A \end{cases}, \end{aligned}$$

which we call *indicator function* of the set A.

We can use the indicator functions not only to represent the elements of $\mathscr{A}(A,B)$ but also for the sets A and B. We want to give a reformulation of (5.1.1) in terms of indicator functions. We can define the indicator function $\mathrm{ind}^{\mathscr{A}(A,B)}$ in the following way:

$$\mathrm{ind}^{\mathscr{A}(A,B)}(I) = \begin{cases} 1 & \text{if } \forall v \in B : \big(\mathrm{ind}^A(v) \leftrightarrow \mathrm{ind}^{\mathrm{proj}^I(A)}(\mathrm{proj}^I(v))\big), \\ 0 & \text{otherwise.} \end{cases} \tag{5.1.3}$$

This tells us essentially that the functions $\mathrm{ind}^A(\cdot)$ and $\mathrm{ind}^{\mathrm{proj}^I(A)}(\mathrm{proj}^I(\cdot))$ need to be equal on the set B if I is a valid solution of the State-Discrimination-Problem. Next, we can use Definition 3.7 in Section 3.2.1 to express $\mathrm{ind}^{\mathrm{proj}^I(A)}$ in terms of $\mathrm{ind}^A(v)$ and I such that we do not need to construct the indicator function of $\mathrm{proj}^I(A)$ from the set $\mathrm{proj}^I(A)$ explicitly.

Lemma 5.1. Let $A \subseteq \{0,1\}^n$, $\emptyset \neq I \subseteq [n]$. The indicator function of $\mathrm{proj}^I(A)$ is $\mathrm{proj}^I(\mathrm{ind}^A)$, i.e. we have $\mathrm{proj}^I(\mathrm{ind}^A) = \mathrm{ind}^{\mathrm{proj}^I(A)}$ (see Definition 3.7).

Proof. We need to show that $\mathrm{proj}^I(\mathrm{ind}^A)$ is the indicator-function of the set $\mathrm{proj}^I(A)$. We have the equivalences

$$\begin{aligned} &\mathrm{proj}^I(y) \in \mathrm{proj}^I(A) = \big\{\mathrm{proj}^I(z) | z \in A\big\} \\ \Leftrightarrow &\exists z \in A : \mathrm{proj}^I(z) = \mathrm{proj}^I(y) \\ \Leftrightarrow &\exists z \in [y]_{\mathrm{proj}^I} : \mathrm{ind}^A(z) \end{aligned} \tag{5.1.4}$$

The latter one corresponds to the definition of $\mathrm{proj}^I(\mathrm{ind}^A)$ (see Definition 3.7). □

Using the above lemma we will show that $\mathrm{ind}^{\mathscr{A}(A,B)} : \{0,1\}^n \to \{0,1\}$ can be defined in the following way:

$$x \mapsto \begin{cases} 1 & \text{if } \forall y \in \{0,1\}^n : \big[\underbrace{\mathrm{ind}^B(y)}_{y \in B} \to \\ & \big(\underbrace{\mathrm{ind}^A(y)}_{y\in A} \leftrightarrow \underbrace{\exists z \in \{0,1\}^n : \mathrm{ind}^A(z)}_{\exists z \in A:} \wedge \underbrace{\forall i \in [n] : x_i \to [z_i \leftrightarrow y_i]}_{\mathrm{proj}^I(z) = \mathrm{proj}^I(y)}\big)\big], \\ 0 & \text{otherwise.} \end{cases} \tag{5.1.5}$$

We prove this in the following proposition.

Proposition 5.1. The indicator function $\mathrm{ind}^{\mathscr{A}(A,B)}$ can be expressed as the function defined in (5.1.5).

Proof. Let $\emptyset \neq I \subseteq [n]$ be a solution of the State-Discrimination-Problem, i.e. it holds $\mathrm{ind}^{\mathscr{A}(A,B)}(0^I) = 1$. Using Lemma 5.1 we have the equivalence:

$$\begin{aligned} & \emptyset \neq I \subseteq [n] \text{ is a solution} \\ \Leftrightarrow & \forall y \in B : \big(\mathrm{ind}^A(y) \leftrightarrow \underbrace{\exists z \in [y]_{\mathrm{proj}^I} : \mathrm{ind}^A(z)}_{=\mathrm{proj}^I(\mathrm{ind}^A)(y)}\big) \end{aligned} \tag{5.1.6}$$

Now the expression $\exists z \in [y]_{\mathrm{proj}^I}$ can be replaced using the following equivalence

$$z \in [y]_{\mathrm{proj}^I} \Leftrightarrow \underbrace{\forall i \in [n] : (0^I)_i \to \overbrace{(z_i \leftrightarrow y_i)}^{z_i=y_i}}_{\mathrm{proj}^I(z)=\mathrm{proj}^I(y)}.$$

Note that we only need to guarantee equality for the components in I to ensure that $\mathrm{proj}^I(z) = \mathrm{proj}^I(y) \Leftrightarrow z \in [y]_{\mathrm{proj}^I}$ is satisfied. Therefore, the expression (5.1.6) is equivalent to

$$\begin{aligned} & \forall y \in \{0,1\}^n : \big[\overbrace{\mathrm{ind}^B(y)}^{y\in B} \to \big(\overbrace{\mathrm{ind}^A(y)}^{y\in A} \leftrightarrow \\ & \exists z \in \{0,1\}^n : \underbrace{\mathrm{ind}^A(z)}_{z\in A} \wedge \underbrace{\forall i \in [n] : (0^I)_i \to (z_i \leftrightarrow y_i)}_{\mathrm{proj}^I(z)=\mathrm{proj}^I(y)}\big)\big]. \end{aligned} \tag{5.1.7}$$

In summary, for any given $\emptyset \neq I \subseteq [n]$, proj^I is a solution to State-Discrimination-Problem if and only if $\mathrm{ind}^{\mathscr{A}(A,B)}(0^I) = 1$ with $\mathrm{ind}^{\mathscr{A}(A,B)}$ defined as follows:

$$x \mapsto \begin{cases} 1 & \text{if } \forall y \in \{0,1\}^n : \big[\underbrace{\mathrm{ind}^B(y)}_{y\in B} \to \big(\underbrace{\mathrm{ind}^A(y)}_{y\in A} \leftrightarrow \underbrace{\exists z \in \{0,1\}^n : \mathrm{ind}^A(z)}_{\exists z\in A:} \wedge \underbrace{\forall i \in [n] : x_i \to [z_i \leftrightarrow y_i]}_{\mathrm{proj}^I(z)=\mathrm{proj}^I(y)}\big)\big], \\ 0 & \text{otherwise.} \end{cases}$$

□

(5.1.5) can be used to compute the solutions of $\mathscr{A}(A,B)$ by finding the solutions of $\mathrm{ind}^{\mathscr{A}(A,B)}(x) = 1$. We used the solver QFUN described in Janota [2017] in the examples in this chapter. But others are also possible (e.g. Benedetti [2005], Zhang and Malik [2002]). Note that we cannot use a conventional SAT solver due to usage of the universal quantifier in (5.1.5) (see the reformulation of (5.1.5) in Remark 5.1).

Remark 5.1. In order to use the solver described in Janota [2017] we need to bring (5.1.5) in the following form.

$$\begin{aligned} & \forall y \in \{0,1\}^n : [\mathrm{ind}^B(y) \to [(\exists z \in \{0,1\}^n : \underbrace{\mathrm{ind}^A(z) \wedge \forall i \in [n] : x_i \to [z_i \leftrightarrow y_i]}_{:=\alpha(x,y,z)}) \to \mathrm{ind}^A(y)]] \\ \Leftrightarrow & \forall y \in \{0,1\}^n : [\neg\mathrm{ind}^B(y) \vee \neg(\exists z \in \{0,1\}^n : \alpha(x,y,z)) \vee \mathrm{ind}^A(y)] \\ \Leftrightarrow & \forall y \in \{0,1\}^n : [\neg\mathrm{ind}^B(y) \vee (\forall z \in \{0,1\}^n : \neg\alpha(x,y,z)) \vee \mathrm{ind}^A(y)] \\ \Leftrightarrow & \forall y \in \{0,1\}^n \forall z \in \{0,1\}^n : \neg\mathrm{ind}^B(y) \vee \neg\alpha(x,y,z) \vee \mathrm{ind}^A(y) \end{aligned}$$

In general due to the complexity problems of this type cannot be solved for larger systems Janota [2017]. For our experiments we considered examples between 10 and 20 components. However, the running times depend largely on the structure of the Boolean equations which can allow to tackle larger systems as well in specific instances Janota [2017].

We also note that in the special case $\mathscr{A}(A, \{0,1\}^n)$ the computation is much more easy. Indeed, here $\mathscr{A}(A, \{0,1\}^n)$ has a unique minimal element given by the essential components of the indicator function ind^A (i.e. the components ind^A depends on.). For completeness we prove this here as well:

Definition 5.2 ([Crama and Hammer, 2011, p. 30, Definition 1.23])**.** For a Boolean function $\varphi : \{0,1\}^n \to \{0,1\}$ we call a component $i \in [n]$ *essential* if and only if the following holds

$$\exists v \in \{0,1\}^n : \varphi(v) \oplus \varphi(v^{\{i\}}). \tag{5.1.8}$$

For a Boolean function $\varphi \in \mathbb{B}(n,1)$ we call $\operatorname{Ess}(\varphi)$ the set of its essential components.

We obtain the following proposition:

Proposition 5.2. For any $\emptyset \neq A \subsetneq \{0,1\}^n$ the unique inclusion-wise minimal set of $\mathscr{A}(A, \{0,1\}^n)$ is $\operatorname{Ess}(\operatorname{ind}^A)$.

Proof. We use the indicator function $\operatorname{ind}^{\mathscr{A}(A,\{0,1\}^n)}$ defined in (5.1.3) of the State-Discrimination-Problem. I.e. $\emptyset \neq J \subseteq [n]$ is a solution to the State-Discrimination-Problem if and only if for all $x \in \{0,1\}^n$ holds:

$$\left(\operatorname{ind}^A(x) \leftrightarrow \exists y \in [x]_{\operatorname{proj}^J} : \operatorname{ind}^A(y)\right)$$

The set of solutions of the State-Discrimination-Problem is then given by:

$$\mathscr{A}(A) := \mathscr{A}(A, \{0,1\}^n) \tag{5.1.9}$$

$$= \left\{\emptyset \neq J \subseteq [n] | \forall x \in \{0,1\}^n : \left(\operatorname{ind}^A(x) \leftrightarrow \exists y \in [x]_{\operatorname{proj}^J} : \operatorname{ind}^A(y)\right)\right\} \tag{5.1.10}$$

Let $\operatorname{Ess}(\operatorname{ind}^A)$ be the set of essential components of ind^A.
$\underline{\operatorname{Ess}(\operatorname{ind}^A) \in \mathscr{A}(A)}$: Assume $\operatorname{Ess}(\operatorname{ind}^A) \notin \mathscr{A}(A)$. Then there must be $x, y \in \{0,1\}^n$ such that $\operatorname{ind}^A(x) \oplus \operatorname{ind}^A(y)$ and $y \in [x]_{\operatorname{proj}^{I(A)}}$ holds[2]. This implies that there is some set $\emptyset \neq J \subseteq [n] \backslash \operatorname{Ess}(\operatorname{ind}^A)$ such that $x^J = y$ and therefore, there must be a state $s \in \{0,1\}^n$ such that $\operatorname{ind}^A(s) \oplus \operatorname{ind}^A(s^{\{j\}})$ with $j \in J$. Therefore, $j \in J \subseteq [n] \backslash \operatorname{Ess}(\operatorname{ind}^A)$ is an essential component. But since $\operatorname{Ess}(\operatorname{ind}^A)$ consists of the essential components, this is a contradiction to the assumption.
$\underline{\forall J \in \mathscr{A}(A) : \operatorname{Ess}(\operatorname{ind}^A) \subseteq J}$: Assume the contradiction, i.e. $\exists i \in \operatorname{Ess}(\operatorname{ind}^A) \bigcap J^c$. Since i is an essential component of ind^A it holds $\exists x \in \{0,1\}^n : \operatorname{ind}^A(x) \oplus \operatorname{ind}^A(x^{\{i\}})$ and since $i \in J^c = [n] \backslash J$ it even holds $x^{\{i\}} \in [x]_{\operatorname{proj}^J}$, i.e.

$$\exists x \in \{0,1\}^n \exists y \in [x]_{\operatorname{proj}^J} : \operatorname{ind}^A(x) \oplus \operatorname{ind}^A(y),$$

which implies that $J \notin \mathscr{A}(A)$. □

[2] The expression $\operatorname{ind}^A(x) \oplus \operatorname{ind}^A(y)$ means that x or y is in A but not both.

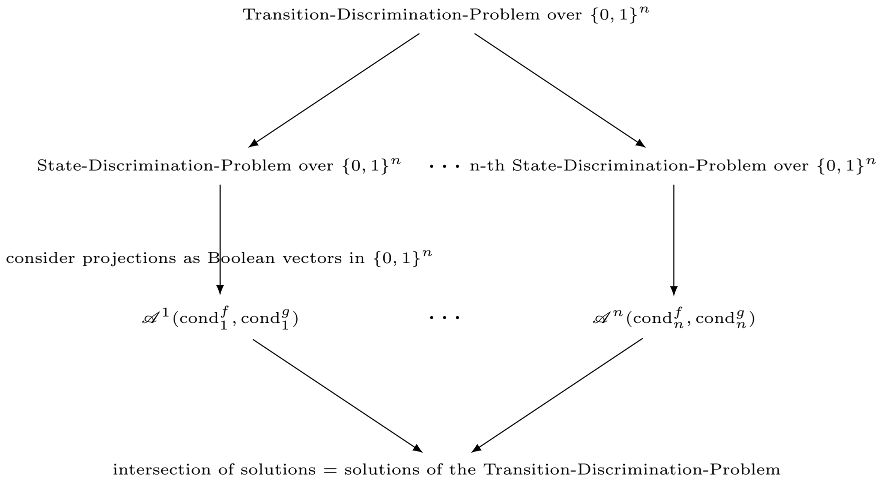

Transition-Discrimination-Problem over $\{0,1\}^n$
State-Discrimination-Problem over $\{0,1\}^n$
· · ·
n-th State-Discrimination-Problem over $\{0,1\}^n$
consider projections as Boolean vectors in $\{0,1\}^n$
$\mathscr{A}^1(\mathrm{cond}_1^f, \mathrm{cond}_1^g)$
· · ·
$\mathscr{A}^n(\mathrm{cond}_n^f, \mathrm{cond}_n^g)$
intersection of solutions = solutions of the Transition-Discrimination-Problem

FIGURE 5.1.4. Scheme of solving the Transition-Discrimination-Problem.

5.1.2. Criteria based on transitions for measuring components. Instead of considering sets of states in $\{0,1\}^n$ in the State-Discrimination-Problem we can also consider sets in $\{0,1\}^n \times \{0,1\}^n$ to select components of a BN we want to measure. Such sets can represent transitions in a Boolean network. Since we can in principle consider elements in $\{0,1\}^n \times \{0,1\}^n$ as elements in $\{0,1\}^{2n}$ we can use the same strategy as in the previous section to find components which allow us to distinguish sets of transitions. If the transitions stem from an ASTG an alternative approach is to split up the problem into maximally n State-Discrimination-Problems. This approach is depicted in Figure 5.1.4. Note, that splitting up the problem into several lower dimensional State-Discrimination-Problems is the preferable strategy, since the complexity of finding the solutions of the State-Discrimination-Problem is very high (see Janota [2017]) and therefore, we need to keep the dimension n of the state space in the State-Discrimination-Problem low. We introduce now the necessary terminology.

To simplify our notation we write for $e = (v, w) \in \{0,1\}^n \times \{0,1\}^n$ instead of $\big(\mathrm{proj}^I(v), \mathrm{proj}^I(w)\big)$ simply $\mathrm{proj}^I(e)$. Completely analogously to the previous section we formulate the following problem:

Problem 5.3 (Transition-Discrimination-Problem)**.** For a given set of transitions $\emptyset \neq E_{\mathrm{async}}(f) \subset E_{\mathrm{async}}(g)$ with $f, g \in \mathbb{B}(n, n)$ we search one or more sets $\mathrm{diff}(E_{\mathrm{async}}(f)) \subseteq I \subseteq [n]$ such that for all $e \in E_{\mathrm{async}}(g)$ it holds

$$e \in E_{\mathrm{async}}(f) \Leftrightarrow \mathrm{proj}^I(e) \in \mathrm{proj}^I(E_{\mathrm{async}}(f)). \tag{5.1.11}$$

In the Transition-Discrimination-Problem we only consider transitions between states with Hamming distance one from each other. Therefore, we assume w.l.o.g. that these sets are the transitions of two ASTGs $G_{\mathrm{async}}(f)$ and $G_{\mathrm{async}}(g)$. In most applications we want to be able to measure the components that change along a transition. Therefore, in the above criterion the restriction $\mathrm{diff}(E_{\mathrm{async}}(f)) \subseteq I$ is added. We denote with $\mathscr{C}(E_{\mathrm{async}}(f), E_{\mathrm{async}}(g))$ the set of solutions of the Transition-Discrimination-Problem. As before we equip the set $\mathscr{C}(E_{\text{async}}(f), E_{\text{async}}(g))$ with the set inclusion as partial ordering. Also the Transition-Discrimination-Problem has in general no unique minimal solution.

To compute the set of solutions $\mathscr{C}(E_{\text{async}}(f), E_{\text{async}}(g))$ we want to split the Transition-Discrimination-Problem into n State-Discrimination-Problems. To do so we consider the functions cond_i^f and cond_i^g, $i \in [n]$ as indicator functions of sets $A_i \subseteq \{0,1\}^n$ and $B_i \subseteq \{0,1\}^n$. More precisely we define

$$\begin{aligned} A_i &:= (\text{cond}_i^f)^{-1}(1), \\ B_i &:= (\text{cond}_i^g)^{-1}(1). \end{aligned}$$

Then we consider the sets of solutions $\mathscr{A}(A_i, B_i)$ of the corresponding State-Discrimination-Problems. To simplify our notation we write $\mathscr{A}(\text{cond}_i^f, \text{cond}_i^g)$ instead of $\mathscr{A}(A_i, B_i)$. In the last section we explained how we can compute the sets $\mathscr{A}(\text{cond}_i^f, \text{cond}_i^g)$. To obtain the set of solutions $\mathscr{C}(E_{\text{async}}(f), E_{\text{async}}(g))$ we just need to intersect all these sets of solutions. We prove this in the following proposition.

Proposition 5.3. Let $\mathscr{C}(E_{\text{async}}(f), E_{\text{async}}(g))$ be the set of solutions of the Transition-Discrimination-Problem and let for $i \in \text{diff}(E_{\text{async}}(f))$ be $\mathscr{A}(\text{cond}_i^f, \text{cond}_i^g)$ be the set of solutions of the State-Discrimination-Problem. Then it holds

$$\mathscr{C}(E_{\text{async}}(f), E_{\text{async}}(g)) = \bigcap_{i \in \text{diff}(E_{\text{async}}(f))} \mathscr{A}(\text{cond}_i^f, \text{cond}_i^g) \cap \{I | \text{diff}(E_{\text{async}}(f)) \subseteq I \subseteq [n]\}.$$

Proof. Let the sets A_i and B_i, $i \in [n]$ be defined as above for $E_{\text{async}}(f)$ and $E_{\text{async}}(g)$.
" $\subseteq$ ": Let $I \in \mathscr{C}(E_{\text{async}}(f), E_{\text{async}}(g))$ be any solution of the Transition-Discrimination-Problem. This means it holds

$$\forall e \in E_{\text{async}}(g) : e \in E_{\text{async}}(f) \Leftrightarrow \text{proj}^I(e) \in \text{proj}^I(E_{\text{async}}(f)). \tag{5.1.12}$$

We want to split up (5.1.12) into n equivalences. Let us define for $i \in [n]$ with $E^i_{\text{async}}(f)$ the set of transitions in $E_{\text{async}}(f)$ where the component i changes, i.e.

$$E^i_{\text{async}}(f) := \{(v, v^{\{i\}}) | (v, v^{\{i\}}) \in E_{\text{async}}(f)\}$$

and analogously for g we define the sets $E^i_{\text{async}}(g)$. It holds

$$E_{\text{async}}(f) = \bigcup_{i \in \text{diff}(E_{\text{async}}(f))} E^i_{\text{async}}(f)$$

$$\Rightarrow \text{proj}^I\big(E_{\text{async}}(f)\big) = \bigcup_{i \in \text{diff}(E_{\text{async}}(f))} \text{proj}^I\big(E^i_{\text{async}}(f)\big) \tag{5.1.13}$$

Clearly, due to $\text{diff}(E_{\text{async}}(f)) \subseteq I$ the sets $E^i_{\text{async}}(f)$ are disjunct from each other. The same holds true for the sets $E^i_{\text{async}}(g)$ with $i \in \text{diff}(E_{\text{async}}(f)) \subseteq I$. Therefore, we can replace (5.1.12) by $|\text{diff}(E_{\text{async}}(f))|$ equivalences

$$\forall e \in E^i_{\text{async}}(g) : e \in E^i_{\text{async}}(f) \Leftrightarrow \text{proj}^I(e) \in \text{proj}^I(E^i_{\text{async}}(f)) \tag{5.1.14}$$

with $i \in \text{diff}(E_{\text{async}}(f))$. Using the sets A_i and B_i we obtain:

$$\forall \underbrace{(v, v^{\{i\}}) \in E^i_{\text{async}}(g)}_{\Leftrightarrow v \in B_i} : \big[\underbrace{(v, v^{\{i\}}) \in E^i_{\text{async}}(f)}_{\Leftrightarrow v \in A_i} \Leftrightarrow \underbrace{(\text{proj}^I(v), \text{proj}^I(v^{\{i\}})) \in \text{proj}^I(E^i_{\text{async}}(f))}_{\Leftrightarrow \text{proj}^I(v) \in \text{proj}^I(A_i)}\big].$$

Consequently, it holds $I \in \mathscr{A}(A_i, B_i) = \mathscr{A}(\mathrm{cond}_i^f, \mathrm{cond}_i^g)$. Obviously, it also holds $\mathrm{diff}(E_{\mathrm{async}}(f)) \subseteq I$.

" $\supseteq$ ": Assume $I \in \bigcap_{i \in \mathrm{diff}(E_{\mathrm{async}}(f))} \mathscr{A}(A_i, B_i)$ and $\mathrm{diff}(E_{\mathrm{async}}(f)) \subseteq I$. For each $i \in \mathrm{diff}(E_{\mathrm{async}}(f))$ we obtain

$$\forall v \in B_i : v \in A_i \Leftrightarrow \mathrm{proj}^I(v) \in \mathrm{proj}^I(A_i).$$

It follows that all $|\mathrm{diff}(E_{\mathrm{async}}(f))|$ equivalences (5.1.14) hold. But then also

$$\forall e \in \bigcup_{i \in \mathrm{diff}(E_{\mathrm{async}}(f))} E^i_{\mathrm{async}}(g) : \Big[e \in \underbrace{\bigcup_{i \in \mathrm{diff}(E_{\mathrm{async}}(f))} E^i_{\mathrm{async}}(f)}_{= E_{\mathrm{async}}(f)} \Leftrightarrow \mathrm{proj}^I(e) \in \underbrace{\bigcup_{i \in \mathrm{diff}(E_{\mathrm{async}}(f))} \mathrm{proj}^I\big(E^i_{\mathrm{async}}(f)\big)}_{= \mathrm{proj}^I(E_{\mathrm{async}}(f))} \Big]$$

holds from which (5.1.12) follows. Therefore, I is an element of the set $\mathscr{C}(E_{\mathrm{async}}(f), E_{\mathrm{async}}(g))$. □

This means in order to find the solutions of the State-Discrimination-Problem we can compute the $|\mathrm{diff}(E_{\mathrm{async}}(f))| \leq n$ sets $\mathscr{A}(\mathrm{cond}_i^f, \mathrm{cond}_i^g)$, $i \in \mathrm{diff}(E_{\mathrm{async}}(f))$.

5.1.3. Showcase: cell-fate decision. We investigate the reduced model in Calzone et al. [2010] modeling cell-fate decision. We want to use the criteria developed in Section 5.1.1 and Section 5.1.2. Attractors in this model correspond to the death or survival of the cell, which can be triggered by certain Cytokines, represented as inputs in the model. Two types of cell deaths are distinguished: apoptosis, a controlled cell death, and a type of non-apoptotic cell death with morphological features of necrosis. Here, we refer to this type of cell death simply as necrosis. Models of this regulatory network and their understanding could help to identify how and under which conditions the cell chooses between different types of cell deaths and survival. For demonstrating the criteria of Section 5.1.1 and Section 5.1.2 we use the reduced model derived from the complete one for the analysis in Calzone et al. [2010]. The corresponding complete model can be found in the BioModels database with the reference MODEL0912180000. The complete model has 27 steady states, which are the only attractors of the model [Calzone et al., 2010, Figure 2]. For the reduction of the model results from Paulevé and Richard [2012], Naldi et al. [2011] are used. The steady states in the reduced model can be attributed one-to-one to steady states in the complete model via a projection. This implies that results on the components which need to be measured can be lifted up to the complete model.

The Boolean function $f : \{0,1\}^{11} \to \{0,1\}^{11}$, representing the reduced model, is defined in Table 5.1.1 and its interaction graph is depicted in Figure 5.1.5.

Alias	Component	Boolean function
x_1	ATP	$\neg MPT$
x_2	$C3$	$ATP \wedge MOMP \wedge \neg NFkB$
x_3	$C8$	$(TNF \vee FAS \vee C3) \wedge \neg NFkB$
x_4	FAS	FAS
x_5	$MOMP$	$MPT \vee (C8 \wedge \neg NFkB)$
x_6	MPT	$ROS \wedge \neg NFkB$
x_7	$NFkB$	$(cIAP \wedge RIP1) \wedge \neg C3$
x_8	$RIP1$	$\neg C8 \wedge (TNF \vee FAS)$
x_9	ROS	$\neg NFkB \wedge (RIP1 \vee MPT)$
x_{10}	TNF	TNF
x_{11}	$cIAP$	$(NFkB \vee cIAP) \wedge \neg MOMP$

TABLE 5.1.1. Reduced model in [Calzone et al., 2010, Table 1]. The components TNF and FAS are inputs.

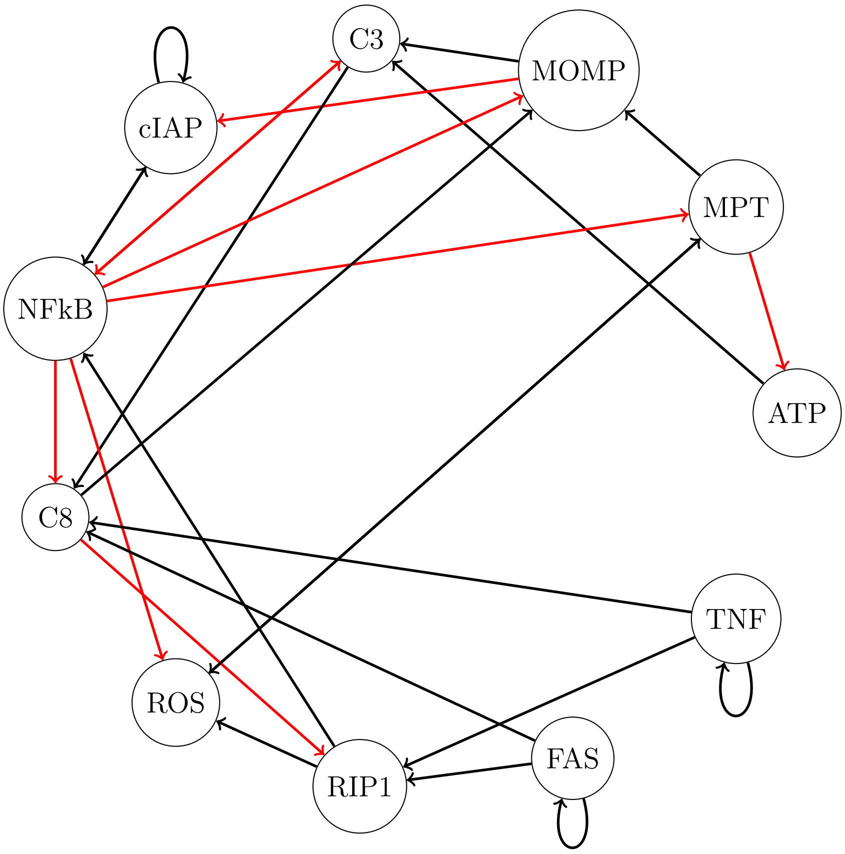


FIGURE 5.1.5. The (global) interaction graph of the Boolean function defined in Table 5.1.1.

Computing the attractors (Definition 1.9) yields 13 steady states:

$$S = \{\underbrace{00001100100}_{s^1}, \underbrace{00101100110}_{s^2}, \underbrace{00111100100}_{s^3}, \underbrace{00111100110}_{s^4}, \underbrace{10000000000}_{s^5},$$
$$\underbrace{10000000001}_{s^6}, \underbrace{10000011011}_{s^7}, \underbrace{10010011001}_{s^8}, \underbrace{10010011011}_{s^9}, \underbrace{11101000000}_{s^{10}},$$
$$\underbrace{11101000010}_{s^{11}}, \underbrace{11111000000}_{s^{12}}, \underbrace{11111000010}_{s^{13}}\}.$$

These 13 steady states are exactly the steady states of the complete model with $FADD = 1$ projected on the components of the reduced model.

According to Calzone et al. [2010] the components $NFkB$ and ATP can be used as markers for the survival, apoptosis or necrosis of the cell. More precisely, we define the indicator functions of these groups in the following way:

$$\begin{aligned}\mathrm{ind}^{\mathrm{survival}} &: x \mapsto x_7,\\ \mathrm{ind}^{\mathrm{necrosis}} &: x \mapsto \neg x_1,\\ \mathrm{ind}^{\mathrm{apoptosis}} &: x \mapsto \neg \mathrm{ind}^{\mathrm{survival}}(x) \wedge \neg \mathrm{ind}^{\mathrm{necrosis}}(x).\end{aligned}$$

Using these markers the 13 steady states can be partitioned into three groups (Table 5.1.2)[3]. We can use the criteria from Section 5.1.1 to ask if the above components are the only possible markers for this classification:

Example 5.3. In Table 5.1.2 we see three groups of steady states and for each of these groups the projections are listed which enable us to decide whether a state belongs to this group or not. Combing any two such projections (i.e. taking the union) from two rows of this table will allow us to distinguish between the three groups. For example in Table 5.1.2 we see that the projection represented by 01001000000 will allow us to distinguish between steady states corresponding to necrosis and steady states not corresponding to necrosis. Components marked with a 1 need to be measured (see (5.1.2) in Section 5.1.1). In the second row we see that the projection represented by 10000001000 allows us to distinguish between apoptotic steady states and the remaining steady states. We conclude that the projection 11001001000 will allow us to distinguish between apoptotic, necrotic, and surviving steady states. Using the numbering in Table 5.1.1 we obtain that the components ATP, $C3$, $MOMP$ and $RIP1$ could be used as markers for the three groups of cell death as well. The indicator function on these components is given by:

$$\begin{aligned}\mathrm{ind}^{\mathrm{survival}} &: x \mapsto x_1 \wedge \neg x_2 \wedge \neg x_5 \wedge x_8,\\ \mathrm{ind}^{\mathrm{necrosis}} &: x \mapsto \neg x_2 \wedge x_5,\\ \mathrm{ind}^{\mathrm{apoptosis}} &: x \mapsto x_1 \wedge \neg x_8.\end{aligned}$$

In contrast to the original indicator function the ones above depend on four components. Nevertheless, such alternative markers could be useful in applications if for example they are more convenient to measure for biologists.

Instead of solely considering the steady states we could as well try to distinguish the strong basins of attractions of the three groups of steady states.

Example 5.4. In Table 5.1.3 we see for each group of attractors (corresponding to necrosis, apoptosis and survival of the cell) the projections which allow us to distinguish between the strong basins of attraction. Combining 00000100000 from the first row and 00010010110 from the second row of Table 5.1.3 allows us for example to distinguish between the strong basins of

[3] The three sets of states $\{x \in \{0,1\}^{11} | \mathrm{ind}^{\mathrm{survival}}(x)\}$, $\{x \in \{0,1\}^{11} | \mathrm{ind}^{\mathrm{necrosis}}(x)\}$ and $\{x \in \{0,1\}^{11} | \mathrm{ind}^{\mathrm{apoptosis}}(x)\}$.

attraction of the apoptotic, necrotic and surviving cells. The corresponding indicator functions are the following:

$$\begin{aligned}\mathrm{ind}^{\text{strong basin of surviving ss}} &: x \mapsto \neg\mathrm{ind}^{\text{strong basin of necrotic ss}}(x) \wedge \neg\mathrm{ind}^{\text{strong basin of apoptotic ss}}(x),\\ \mathrm{ind}^{\text{strong basin of necrotic ss}} &: x \mapsto x_6,\\ \mathrm{ind}^{\text{strong basin of apoptotic ss}} &: x \mapsto \left[\neg x_4 \wedge \neg x_9 \wedge \neg x_{10}\right] \vee \left[\neg x_7 \wedge \neg x_9\right].\end{aligned}$$

Other combinations of projections are of course possible, too.

Sometimes we are interested in which “moment” a trajectory enters a strong basin of attraction. This means we want to distinguish between the strong and weak basin of a group of attractors. In our case, states in the strong basin lie on trajectories leading to steady states corresponding to one specific type of cell death, while states in the weak basin lie on trajectories which can lead to steady states corresponding to different types of cell death. The components necessary for detecting such a “decision” could elucidate possible ways of interventions.

Example 5.5. In Table 5.1.4 the weak and strong basins for each group of attractors are listed. In Table 5.1.5 the projections are listed necessary to distinguish for each of them between its weak and strong basin of attraction. We see that in all three cases it is necessary to measure almost all components of the network. For finding these projections we make the assumption that we only observe states in the strong basins.

To reduce the amount of components, we could instead focus on the transitions from the weak to the strong basin and consider only transition where specific components change.

Example 5.6. Assume we are interested in the transitions from the weak to the strong basin of attraction for the attractors corresponding to the necrotic steady states. Due to the complex form of the basins we would need to measure in this case almost all components. However, we could focus on the changes of specific components and ask what we need to measure to detect all transitions $(v, v^{\{i\}})$, $i \in [11]$ leading from the weak to the strong basin of attraction. For most components this leads still to many components we need to measure. But for example for the component cIAP we obtain the minimal projection 00000110100. I.e. in this case we only need to measure four components: cIAP (corresponding to x_{11})[4] and the three other components as indicated by the minimal projection (MPT, $NFkB$ and ROS).

More precisely: For the set A in the State-Discrimination-Problem the indicator function is given by[5]

$$\mathrm{ind}^{\text{weak basin of necrotic steady states}}(v) \wedge \mathrm{ind}^{\text{strong basin of necrotic steady states}}(v^{\{11\}}) \wedge \mathrm{cond}_{11}^{f}(v).$$

[4] Since this is the component that changes.

[5] We need to restrict the condition function of f in the 11th component cond_{11}^{f} to transitions from the weak basin to the strong basin. The term $\mathrm{ind}^{\text{weak basin of necrotic steady states}}(v)$ is only satisfied if the origin v of the transition $(v, v^{\{11\}})$ is in the weak basin, $\mathrm{ind}^{\text{strong basin of necrotic steady states}}(v^{\{11\}})$ is only satisfied if the destination of the transition $v^{\{11\}}$ is inside the strong basin. Combining these three conditions we see that $\mathrm{ind}^{\text{weak basin of necrotic steady states}}(v) \wedge \mathrm{ind}^{\text{strong basin of necrotic steady states}}(v^{\{11\}}) \wedge \mathrm{cond}_{11}^{f}(v)$ is the necessary and sufficient condition for such transitions.

This indicator function takes the value one on the subspace $****110*1*1$ (This means $A = ****110*1*1$). The indicator function of the set B in the State-Discrimination-Problem is given by $\text{cond}^f_{11}(v)$ and it can be shown that B is the union of the subspaces $****0*1***0$ and $****1*****1$. Computing the minimal solutions of the State-Discrimination-Problem with this choice of sets yields the single minimal projection represented by 00000110100. Furthermore, it is possible to show that for any change in the 11th component cIAP we could evaluate the term $x_6 \wedge \neg x_7 \wedge x_9$ derived from the projection and the subspace representation[6] to check whether a change in the value of cIAP causes the "trajectory" to lead towards a necrotic steady state.

5.1.4. An algebraic reformulation of the State-Discrimination-Problem.

In this section we demonstrate that the criteria in Section 5.1.1 and Section 5.1.2 can also be considered from an algebraic point of view. This is possible since we can consider Boolean functions $\mathbb{B}(n,1)$ as elements of the polynomial ring $\mathbb{F}_2[x_1,\dots,x_n]/\langle x_1^2 - x_1, \dots, x_n^2 - x_n\rangle$. Then the idea is to consider instead of the set $B \subseteq \{0,1\}^n$ the *vanishing ideal* $\mathcal{I}(B)$ of B. Due to the Boolean Strong Nullstellensatz there is an easy relation between the set B and its vanishing ideal $\mathcal{I}(B)$. Instead of considering Boolean functions which allow us to distinguish the set A and its complement in the State-Discrimination-Problem we can consider the equivalence class $\text{ind}^A/\mathcal{I}(B)$. The first result of this section is that the solution diagram $\mathscr{A}(A,B)$ is actually given by the set of variables of the polynomials in the equivalence class $\text{ind}^A/\mathcal{I}(B)$ (Proposition 5.4). Afterwards we consider how we can use Gröbner bases to obtain minimal solutions of $\mathscr{A}(A,B)$.

For a more extensive introduction to algebraic geometry and Gröbner bases we refer to Cox et al. [2007]. We use in this section the letter $\mathcal{I}$ for an ideal in $k[x_1,\dots,x_n]$, where k is any field. With $\mathcal{V}(\mathcal{I})$ we denote the set of points for which all polynomials in $\mathcal{I}$ vanish. For a set $S \subseteq k^n$ we denote with $\mathcal{I}(S)$ the vanishing ideal, the set of all polynomials that vanish on S. In the State-Discrimination-Problem we were given two sets $\emptyset \neq A \subset B \subseteq \mathbb{F}_2^n$. In order to give an algebraic formulation of the State-Discrimination-Problem we need to look at the vanishing ideals of the set B. If we consider polynomials in $\mathbb{F}_2[x_1,\dots,x_n]$ we need to take into account that the set of polynomials does not coincide with the set of Boolean functions. This is the case since the polynomials $x_1^2 - x_1, \dots, x_n^2 - x_n$ evaluate to zero on $\mathbb{F}_2^n$. We call the polynomials $x_1^2 - x_1, \dots, x_n^2 - x_n$ *field polynomials*. However, we can say that any two polynomials whose difference is a sum of field polynomials corresponds to the same Boolean function (see e.g. Cheng and Qi [2009]). In other words we can identify the ring of Boolean functions $\mathbb{B}(n,1)$ with $\mathbb{F}_2[x_1,\dots,x_n]/\langle x_1^2 - x_1, \dots, x_n^2 - x_n\rangle$. We will denote both objects with $\mathbb{B}(n,1)$.

In $\mathbb{B}(n,1)$ we have for any set $S \subseteq \mathbb{F}_2^n$ the identity $\mathcal{V}(\mathcal{I}(S)) = S$ due to the *Boolean Strong Nullstellensatz*:

Theorem 5.1 (Boolean Strong Nullstellensatz)**.** (Sato et al. [2011], Gao et al. [2011]) Let $\mathcal{I}$ be an ideal in $\mathbb{B}(n,1)$ such that $\mathcal{V}(\mathcal{I}) \neq \emptyset$. Then for any

[6] The subspace representation corresponds to a representation as a DNF and can be used to obtain the projected indicator function (see Definition 3.7 and Example 3.5).

Group of steady states	Steady states	corresponding minimal projections
Nekrosis	00001100100, 00101100110, 00111100100, 00111100110	01001000000, 10000000000, 00000100000, 00000000100
Apoptosis	10000000000, 10000000001, 11101000000, 11101000010, 11111000000, 11111000010	10000001000, 10000010000, 00010100011, 01010100010, 01001010000, 01001001000, 00011100010, 00011000110, 00000010100, 00000001100, 00000110000, 00000101000, 01011000010, 00110100010, 00010000111, 00110000110, 10010000011, 10110000010, 10011000010, 11010000010, 01010000110
Survival	10000011011, 10010011001, 10010011011	00110000010, 11010000010, 01010000110, 01010100010, 00010000011, 00011000010, 00000001000, 00000010000

Table 5.1.2. Three groups of steady states (red: necrotic, green: apoptotic, blue: survival) and the projections allowing to distinguish one group from the other two.

polynomial $h(x) \in \mathbb{B}(n,1)$ it holds

$$h(x) \in \mathcal{I} \Leftrightarrow \forall v \in \mathcal{V}(\mathcal{I}) : h(v) = 0.$$

Let us now come back to the problem of minimal sets of components we need to measure. We assume the sets A and B in the State-Discrimination-Problem to be represented by the indicator functions ind^A and ind^B. The implicit representation of the sets A and B can have advantages. For example if the set B consists of the steady states of a synchronous or asynchronous

	Projections to distinguish strong basins of attractions	Projections to distinguish weak basins of attractions
Necrosis	00000100000, 00000010100, 00000001101, 00110000111	01111111111
Apoptosis	00010010110, 00110100010, 00010101010, 00010001110, 00010110010, 00110000110	01111111111
Survival	00010010010, 00110000011, 00010101010, 00110100010, 00010001011	00110011011

TABLE 5.1.3. Minimal projections enabling to distinguish one strong basin of attraction from the two other.

BN represented by a function $f \in \mathbb{B}(n,n)$, then $\text{ind}^B(x)$ can be represented by the Boolean formula $\underbrace{\left[\bigwedge_{i\in[n]} (f_i(x) \leftrightarrow x_i)\right]}_{\Leftrightarrow f(x)=x}$ without need for explicit computation of the steady states. We define the ideal

$$\mathcal{I} := \langle \text{ind}^B(x) - 1 \rangle \subseteq \mathbb{B}(n,1).$$

In other words $\mathcal{I}$ is the vanishing ideal of B. We have the following relation to the State-Discrimination-Problem.

Proposition 5.4. Let $\varphi \in \mathbb{B}(n,1)$ be any polynomial such that $\text{ind}^A - \varphi \in I$ holds. Let J be the essential components of φ, i.e. $J := \text{Ess}(\varphi)$. Then it holds $J \in \mathscr{A}(A,B)$ (I.e. J is a solution to the State-Discrimination-Problem).

PROOF. Due to the Boolean Strong Nullstellensatz $\text{ind}^A - \varphi \in \mathcal{I}$ means that ind^A and φ are equal on the set B.[7] In order to know the value of φ we only need to know the values of its essential components. Therefore, the essential components of φ are enough to decide if a state in B is in A or not. □

On the other hand every solution $J \subseteq [n]$ of the State-Discrimination-Problem corresponds to a polynomial depending on the variables with indices in J in the equivalence class $\text{ind}^A/\mathcal{I}$:

[7] It holds $V(\mathcal{I}) = B$ and according to the Boolean Strong Nullstellensatz $\forall v \in B : \varphi(v) - \text{ind}^A(v) = 0 \Leftrightarrow \varphi(v) - \text{ind}^A(v) \in \mathcal{I}$

Marker	Steady states	weak basin of attraction	strong basin of attraction
Necrosis	00001100100, 00101100110, 00111100100, 00111100110	******1**10, ******10*1*, ****1*1**1*, ***0****10*, ***1**1***0, ***1**10***, ***11*1****, **0******10, **0****0*1*, **0***0**1*, **0*1****1*, **01******0, **01***0***, **01**0****, **011******, **1****1*1*, **1***1**1*, **11***1***, **11**1****, *1****1**1*, *1*1**1****, *10******1*, *101*******, ********1*0, *******01**, *******1**0, ******0*1**, ******01***, ****1***1**, ****1**1***, **1*****1**, *1******1**, *1*****1***, *****1*****	***0*10010*, **1**1001**, *****10*1*0
Apoptosis	10000000000, 10000000001, 11101000000, 11101000010, 11111000000, 11111000010	***0***1*01, **0****0*11, **0***0**11, **0**1***11, **0*1****11, **01***0**1, **01**0***1, **01*1****1, **011*****1, *10******11, *101******1, ******01**1, *****1*1**1, ****1**1**1, ***0****00*, ***0**1**0*, ***0*0***0*, **1****1**1, *1*****1**1, ********0*0, *******00**, ******0*0**, ******1***0, ******10***, *****0****0, *****0*0***, *****00****, *****1**0**, *****11****, ****1***0**, ****1*1****, ****10*****, **1*****0**, **1***1****, **1**0*****, *1******0**, *1****1****, *1***0*****	*00000*0001, *000000000*, *000001*001, 111*10000**, **1**00001*, **11*0000**
Survival	10000011011, 10010011001, 10010011011	*******1*11, ***1***1**1, **0******11, **01******1, ******1**1*, ***1**1****	*00*0011*11, *0010011**1

TABLE 5.1.4. Weak and strong basins of attractions of the three groups of attractors represented as list of subspaces.

	Projections to distinguish strong and weak basins of attractions
Necrosis	00110111111
Apoptosis	11111111111
Survival	01101111001

TABLE 5.1.5. Minimal components to distinguish between the strong and weak basin of attraction of each group of attractors.

Proposition 5.5. Let J be any solution of the State-Discrimination-Problem, i.e. $J \in \mathscr{A}(A, B)$. Then it holds

$$\mathrm{ind}^A - \mathrm{ind}^{\mathrm{proj}^J(A)} \in \mathcal{I},$$

where we interpret and $\mathrm{ind}^{\mathrm{proj}^J(A)}$ as a polynomial in $\mathbb{B}(n, 1)$. Furthermore, for all $v \in B$ it holds:

$$\mathrm{ind}^A(v) = \mathrm{ind}^{\mathrm{proj}^J(A)}(\mathrm{proj}^J(v)).$$

PROOF. Since J is a solution to State-Discrimination-Problem it holds for all $v \in B$ that

$$v \in A \Leftrightarrow \mathrm{proj}^J(v) \in \mathrm{proj}^J(A)$$
$$\Leftrightarrow \mathrm{ind}^A(v) = \mathrm{ind}^{\mathrm{proj}^J(A)}(\mathrm{proj}^J(v)),$$

which proves the second claim. According to the Boolean Strong Nullstellensatz it follows $\forall v \in B : \mathrm{ind}^{\mathrm{proj}^J(A)}(\mathrm{proj}^J(v)) - \mathrm{ind}^A(v) = 0 \Leftrightarrow \mathrm{ind}^{\mathrm{proj}^J(A)} - \mathrm{ind}^A \in \mathcal{I}$. □

Proposition 5.4 and Proposition 5.5 allow us to describe the set of solutions $\mathscr{A}(A, B)$ of the State-Discrimination-Problem and the corresponding indicator functions of the projected sets as equivalence classes with respect to the vanishing ideal of B.

We want to obtain inclusion-wise minimal elements of the solution-diagram $\mathscr{A}(A, B)$ of the State-Discrimination-Problem. For this purpose we introduce the notions of a *monomial ordering* on the monomials of $k[x_1, \ldots, x_n]$. They are used in the context of Gröbner bases which will be introduced in a moment. We define these monomial orderings on $\mathbb{Z}^n_{\geq 0}$.

Definition 5.3. [Cox et al., 2007, p. 69]A monomial ordering on the set of monomials in $k[x_1, \ldots, x_n]$ is any relation $\succ$ on $\mathbb{Z}^n_{\geq 0}$ or equivalently, any relation on the set of monomials x^α, $\alpha \in \mathbb{Z}^n_{\geq 0}$ satisfying:
(i) $\succ$ is a total ordering on $\mathbb{Z}^n_{\geq 0}$.
(ii) If $\alpha \succ \beta$ and $\gamma \in \mathbb{Z}^n_{\geq 0}$, then $\alpha + \gamma \succ \beta + \gamma$.
(iii) $\succ$ is a well-ordering on $\mathbb{Z}^n_{\geq 0}$ (Every nonempty subsets of $\mathbb{Z}^n_{\geq 0}$ has a smallest element under $\succ$.).

Each monomial can be identified with a point in $\mathbb{Z}^n_{\geq 0}$ depending on which variable occurs with which power in the monomial. For example the monomial $x_1 \cdot x_3$ in $\mathbb{F}_2[x_1, x_2, x_3]$ is identified with 101, since it depends on x_1^1 and x_3^1. For $\alpha \in \mathbb{Z}^n_{\geq 0}$ we write $x^\alpha := \prod_{i \in [n]} x_i^{\alpha_i}$. For any polynomial in $p \in k[x_1, \ldots, x_n]$ and monomial ordering $\succ$ we write $LT(p)$ for the largest term occurring in p with respect to $\succ$. Here, we are only interested in specific orderings – the lexicographical orderings – on these monomials.

Definition 5.4. [Cox et al., 2007, p. 70]Let $\alpha = \begin{pmatrix}\alpha_1 & \dots & \alpha_n\end{pmatrix}$ and $\beta = \begin{pmatrix}\beta_1 & \dots & \beta_n\end{pmatrix}$ be two elements in $\mathbb{Z}^n_{\geq 0}$. We say $\alpha \succ_{lex} \beta$ if the leftmost nonzero entry of $\alpha - \beta \in \mathbb{Z}^n$ is positive. We write $x^\alpha \succ_{lex} x^\beta$ if $\alpha \succ_{lex} \beta$.

This gives us a way to order the minimal solutions of the State-Discrimination-Problem.

Next, we explain how we can obtain the minimal solutions of $\mathscr{A}(A, B)$. For this purpose we need the notion of Gröbner bases. Let $\prec$ be any term ordering.

Definition 5.5 ([Sturmfels, 1996, p. 1])**.** For an ideal $\mathcal{I} \subseteq k[x]$ we define its *initial ideal* as the ideal

$$in_\prec(\mathcal{I}) := \langle in_\prec(f) | f \in \mathcal{I}\rangle,$$

where $in_\prec(f)$ denotes the *initial monomial* of f. A finite subset $G \subseteq \mathcal{I}$ is a *Gröbner basis* for $\mathcal{I}$ with respect to $\prec$ if $in_\prec(\mathcal{I})$ is generated by $\{in_\prec(g) | g \in G\}$. If no element of the Gröbner basis G is redundant, then G is *minimal*. It is called *reduced* if for any two distinct elements $g, g' \in G$ no term of g' is divisible by $in_\prec(g)$. The reduced Gröbner basis is unique for an ideal and a term ordering and we denote it by $G_\prec(\mathcal{I})$, provided the leading coefficients of the elements of G are set to one. Every monomial not lying in $in_\prec(\mathcal{I})$ is called *standard monomial*.

We have the following proposition explaining the outcome of the division algorithm (see [Cox et al., 2007, p. 61] for details about the division algorithm).

Proposition 5.6 ([Sturmfels, 1996, p. 1],[Cox et al., 2007, p. 82, Prop. 1])**.** The standard monomials form a k-vector space basis for the residue ring $k[x]/\mathcal{I}$.

The division algorithm rewrites every polynomial f modulo $\mathcal{I}$ uniquely as a linear combination of these standard monomials [Sturmfels, 1996, p. 1].

Assume $\prec$ is a a lexicographical ordering. Let $G_\prec(\mathcal{I})$ be the reduced Gröbner basis for I with respect to $\prec$. Let φ be the the result of the division algorithm applied to ind^A (i.e. it holds $\varphi - \mathrm{ind}^A \in \mathcal{I}$.) with $G_\prec(\mathcal{I})$ and the ordering $\prec$. It would be tempting to conclude due to Proposition 5.4 that the essential components of φ correspond to a minimal element in $\mathscr{A}(A, B)$. However, we need to be careful here. The essential components of φ correspond not necessarily to a minimal solution of the State-Discrimination-Problem. To see this consider the following counter example.

Example 5.7. Consider the set $A = \{111\}$ and $B = \{000, 111, 011, 101\}$ in the State-Discrimination-Problem. We can express the set B as the set of vanishing points of an ideal generated by $x_1x_2 + x_1 + x_2 + x_3$, i.e. $B = V(\langle x_1x_2 + x_1 + x_2 + x_3\rangle)$. It is easy to see that $\{1, 2\} \in \mathscr{A}(A, B)$ is the only minimal element in $\mathscr{A}(A, B)$ with corresponding indicator function $\mathrm{ind}^{\mathrm{proj}^{\{1,2\}}(A)}(x) = x_1x_2$. When using the lexicographical ordering with $x_1 \succ x_2 \succ x_3$ and applying the division algorithm with a corresponding

Gröbner basis (In this case this is just $\{x_1x_2 + x_1 + x_2 + x_3\}$) we obtain $\varphi(x) = x_1 + x_2 + x_3$. We have $\{1,2\} \subset \mathrm{Ess}(\varphi) = \{1,2,3\}$. But $\{1,2,3\}$ is not a minimal solution of the State-Discrimination-Problem. This second representative φ depends on more variables than $\mathrm{ind}^{\mathrm{proj}^{\{1,2\}}(A)}$. However, its leading monomial x_1 has a smaller degree than x_1x_2.

However, we can show that for some lexicographical ordering any minimal element of $\mathscr{A}(A,B)$ is the result of the reduction algorithm with corresponding Gröbner basis. This is achieved by choosing for a minimal solution $J \in \mathscr{A}(A,B)$ a lexicographical ordering such for all $i \in [n]\backslash J$ and $j \in J$ it holds $x_i \succ x_j$.

Proposition 5.7. Let J be any minimal solution in $\mathscr{A}(A,B)$ then there is a lexicographical ordering $\prec$ such that for the result of the division algorithm – let us say φ – with respect to this ordering and a Gröbner basis $G_\prec(\mathcal{I})$ it holds

$$\mathrm{Ess}(\varphi) = J.$$

Proof. Choose a lexicographical ordering $\prec$ such that for all $i \in [n]\backslash J$ and $j \in J$ it holds $x_i \succ x_j$ holds. W.l.o.g. we can say that this lexicographical ordering is given by $x_1 \succ \cdots \succ x_n$ and that $j_{\min}$ is the minimal element in J, i.e. we have $J = \{j_{\min}, \ldots, n\}$. Let us say φ is the result of the division algorithm with respect to the corresponding Gröbner basis applied to ind^A. Now assume $\mathrm{Ess}(\varphi) \neq J$. It cannot hold $\mathrm{Ess}(\varphi) \subset J$, since according to Proposition 5.4 then J would not be a minimal solution due to $\mathrm{Ess}(\varphi) \in \mathscr{A}(A,B)$.
Consequently, there is some smallest element j in $\mathrm{Ess}(\varphi) \bigcap [n]\backslash J$. But $x_j = LT(\varphi) = LT(\varphi - \mathrm{ind}^{\mathrm{proj}^J(A)})$ and due to Proposition 5.5 we have $\varphi - \mathrm{ind}^{\mathrm{proj}^J(A)} \in \mathcal{I}$. This shows that $LT(\varphi)$ is not a standard monomial which is a contradiction due to Proposition 5.6. □

So in order to find the minimal solutions of $\mathscr{A}(A,B)$ we can iterate over the lexicographical orderings. This is computationally very costly. However, one can either focus on a specific subset of lexicographic orderings interesting for applications or develop methods taking the structure of the Gröbner basis into account to avoid enumerating all orderings. We illustrate this in the following example.

Example 5.8. Let us continue with Example 5.7. In the following table we see the six possible lexicographical orderings and the corresponding solutions in $\mathscr{A}(A,B)$.

lex. ordering	Gröbner basis	Result of Reduction Algorithm	Corresponding solution in $\mathscr{A}(A,B)$
1,2,3	$x_1x_2 + x_1 + x_2 + x_3$	$x_1+x_2+x_3$	$\{1,2,3\}$
1,3,2	$x_1x_2 + x_1 + x_3 + x_2$	$x_1+x_3+x_2$	$\{1,2,3\}$
2,1,3	$x_2x_1 + x_2 + x_1 + x_3$	$x_2+x_1+x_3$	$\{1,2,3\}$
2,3,1	$x_2x_1 + x_2 + x_3 + x_1$	$x_2+x_3+x_1$	$\{1,2,3\}$
3,1,2	$x_3 + x_1x_2 + x_1 + x_2$	x_1x_2	$\{1,2\}$
3,2,1	$x_3 + x_2x_1 + x_2 + x_1$	x_2x_1	$\{1,2\}$

We can see that indeed for the lexicographical ordering $x_3 \succ x_1 \succ x_2$ and $x_3 \succ x_2 \succ x_1$ the reduction algorithm finds the minimal solution of the State-Discrimination-Problem. Furthermore, we see that the first four lexicographical orderings produce a Gröbner basis with identical initial ideal. Therefore, the outcome of the division algorithm will be in these cases always the same.

5.2. Distinguishing Boolean models with criteria based on walks

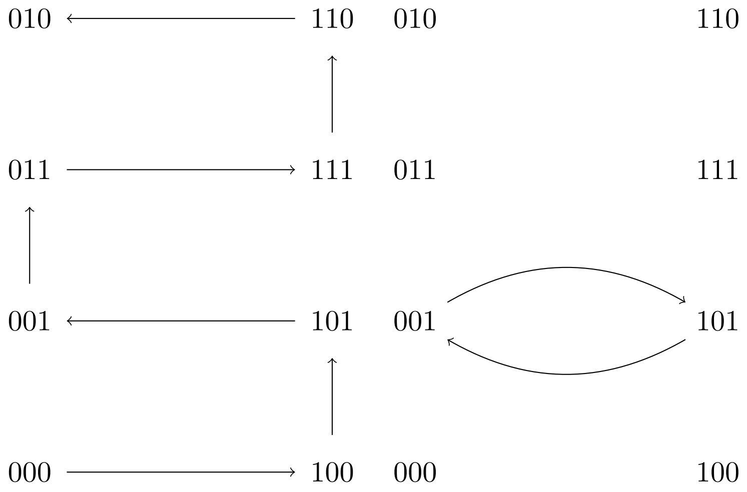
In the previous section we investigated criteria to distinguish different ASTGs based on a limited number of components we measure. One of the criteria was based on the transitions of the ASTGs. However, in many scenarios such criteria will have limited use since they require many states to be measured. However, if we focus not solely on transitions, but walks in ASTGs with potentially longer lengths in certain cases we might be still able to distinguish two models projected on a set of components even though their transitions are after the projection identical. To find these walks we will follow here an approach based on finite automata. In the context of Boolean networks they have been previously used for so-called fixable families Gadouleau and Richard [2018] and for Boolean control networks Zhang and Zhang [2016]. Büchi automata have been used for model checking of BNs (see e.g. Klarner et al. [2012], Streck and Siebert [2015]). The problem we will consider here is also related to observability of Boolean control networks mostly investigated in the context of synchronous Boolean networks Cheng and Qi [2009], Fornasini et al. [2013]. Before we start, we look at a simple example to demonstrate how walks of length bigger or equal than two can be used to distinguish between models.

Example 5.9. Consider two models: The first model consists of the transitions: $\begin{pmatrix}0\\0\end{pmatrix} \to \begin{pmatrix}0\\1\end{pmatrix}$ and $\begin{pmatrix}1\\1\end{pmatrix} \to \begin{pmatrix}1\\0\end{pmatrix}$. The second model consists of the transitions: $\begin{pmatrix}0\\0\end{pmatrix} \to \begin{pmatrix}0\\1\end{pmatrix}$ and $\begin{pmatrix}0\\1\end{pmatrix} \to \begin{pmatrix}0\\0\end{pmatrix}$.[8] When projecting to the components $\{2\}$ we obtain in both models the transitions $\begin{pmatrix}0\end{pmatrix} \to \begin{pmatrix}1\end{pmatrix}$ and $\begin{pmatrix}1\end{pmatrix} \to \begin{pmatrix}0\end{pmatrix}$. So according to the criterion in the State-Discrimination-Problem we cannot distinguish the two models. However, if we look at walks with length two then the walk $\begin{pmatrix}0\end{pmatrix} \to \begin{pmatrix}1\end{pmatrix} \to \begin{pmatrix}0\end{pmatrix}$ can occur only in the second model.

5.2.1. Formalization. In order to formalize this idea, we introduce the notion of projected walks.

Definition 5.6. Assume $\emptyset \neq I \subseteq [n]$ and let $p = (s^1, \dots, s^k)$ for $k \in \mathbb{N}$, $s^i \in \{0,1\}^n$, $i \in [k]$ be a walk in an arbitrary ASTG in $\{0,1\}^n$, we call the

[8] In other words, the first model is induced by the function $f : x \mapsto \begin{cases} 01 & \text{if } x = 00 \\ 10 & \text{if } x = 11 \\ x & \text{otherwise} \end{cases}$ and the second model by $g : x \mapsto \begin{cases} 01 & \text{if } x = 00 \\ 10 & \text{if } x = 01 \\ x & \text{otherwise} \end{cases}$.



(A) Assume f is defined by the above ASTG. The corresponding NFA-ϵ Auto$(f, \{1\})$ will accept the sequence 01010 and its subsequences.

(B) Assume g is defined by the above ASTG. The corresponding NFA-ϵ Auto$(g, \{1\})$ will accept an arbitrary long alternating sequence of 0s and 1s.

FIGURE 5.2.1. The smallest word that is accepted by Auto$(g, \{1\})$ but not by Auto$(f, \{1\})$ is the word 10101. Therefore, to distinguish the two corresponding NFAs you need to consider words of length ≥ 5.

projected walk $\text{proj}^I(p)$ the walk obtained of the sequence $(\text{proj}^I(s^1), \ldots, \text{proj}^I(s^k))$ by deleting subsequent identical elements.

For example for $I = \{1\}$ and $p = (000, 010, 110)$ it holds $\text{proj}^I(p) = (0, 1)$. The projected walk $\text{proj}^I(p)$ consists only of two states, since the first two states in the walk p are both mapped to 0. We use the notion of a projected walk to state the following problem:

Problem 5.4. For two ASTGs $G_{\text{async}}(f)$ and $G_{\text{async}}(g)$ induced by functions $f : \{0,1\}^n \to \{0,1\}^n$ and $g : \{0,1\}^n \to \{0,1\}^n$ and a set of components $\emptyset \neq I \subseteq [n]$, what sequences of states do we need to measure to distinguish the two models? More precisely, what are the sequences $\tilde{p} = (\tilde{s}^1, \ldots, \tilde{s}^{\tilde{k}})$, $\tilde{k} \in \mathbb{N}$ of states in $\{0,1\}^{|I|}$ such that there is a walk $p = (s^1, \ldots, s^k)$, $k \in \mathbb{N}$ in $G_{\text{async}}(f)$ such that $\text{proj}^I(p) = \tilde{p}$ but there is no walk in $G_{\text{async}}(g)$ which is projected to $\tilde{p}$ under proj^I?

In order to solve this problem algorithmically we introduce the notion of a nondeterministic finite automaton with $\epsilon-$moves (NFA-ϵ):

Definition 5.7. A *nondeterministic finite automaton with ϵ-moves* (*NFA-ϵ*) is represented formally by a $5 - \textit{tuple}$ $M = (Q, \Sigma, \triangle, q_0, F)$, where

1) Q is a finite set of states.
2) Σ is a finite set of input symbols with $\epsilon \notin \Sigma$.
3) $\triangle$ is a transition function of the form $\triangle : Q \times (\Sigma \cup \{\epsilon\}) \to \mathscr{P}(Q)$.
4) $q_0 \in Q$ is an initial state.
5) $F \subseteq Q$ is the set of final states.

The set Σ is called *alphabet* and its elements $\sigma \in \Sigma$ are called *letters*. Finite sequences over $\Sigma \cup \{\epsilon\}$ are called *words*. The symbol ϵ is called the *empty string*.

We can represent a NFA-ϵ as a transition graph:

Definition 5.8. The transition graph $G = (V, E, W)$ of a NFA-ϵ denoted by $(Q, \Sigma, \triangle, q_0, F)$, where V is the vertex set and E the edge set is defined by:

$$\begin{aligned} V =& Q, \\ E =& \big\{(s,t,a) \in V \times V \times (\Sigma \cup \{\epsilon\}) | t \in \triangle(s,a)\big\}. \end{aligned}$$

We call a word $\tilde{w}$ over the alphabet $\Sigma \cup \{\epsilon\}$ an ϵ-*extensions* of a word w over the alphabet Σ if w can be obtained from $\tilde{w}$ by deleting all occurrences of ϵ. We say a word $w = a_1 \dots a_n$ over an alphabet Σ is accepted by the NFA-ϵ M if some $\epsilon-$extension of w labels a path in the transition graph corresponding to M.

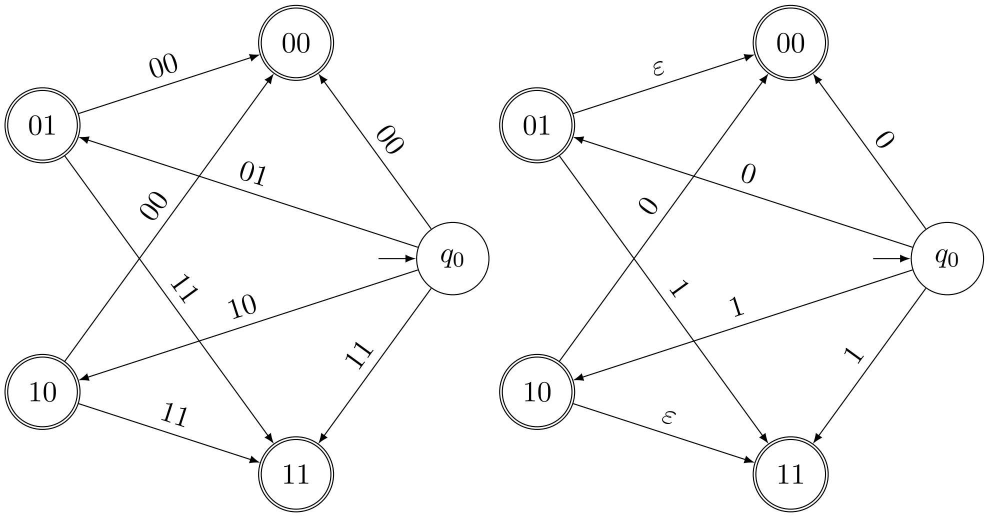
To use tools and theory developed for NFAs, we want to reformulate Problem 5.4 in terms of NFAs. To do so, we associate to an ASTG $G_{\text{async}}(f)$, induced by a Boolean function f and a set of components $\emptyset \neq I \subseteq [n]$, a NFA-$\epsilon$ $\text{Auto}(f, I)$. The words accepted by $\text{Auto}(f, I)$ will be the projected walks of $G_{\text{async}}(f)$ onto the components I. The alphabet of the NFA-ϵ $\text{Auto}(f, I)$ will consist of the projected state space (i.e. $\{0,1\}^{|I|}$).

Definition 5.9. For an ASTG induced by a Boolean function $f : \{0,1\}^n \to \{0,1\}^n$ and a set of components $\emptyset \neq I \subseteq [n]$ we define the following NFA-ϵ with initial state q_0 and a distinguished state q_{failed} denoted by $\text{Auto}(f, I)$:
1) $Q = \{q_0\} \cup \{0,1\}^n \cup \{q_{\text{failed}}\}$ and $F = Q \backslash \{q_0, q_{\text{failed}}\}$.
2) $\Sigma = \{0,1\}^{|I|} = \text{proj}^I(\{0,1\}^n)$
3) $\triangle(q_0, \sigma) = \{t \in F | \text{proj}^I(t) = \sigma\}$ for $\sigma \in \Sigma$.
4) $\triangle(q_{\text{failed}}, \sigma) = \{q_{\text{failed}}\}$ for all $\sigma \in \Sigma$.
5) And for $s \in F$, $\sigma \in \Sigma \cup \{\epsilon\}$ we define:

$$A(s,\sigma) = \begin{cases} \{t \in F | (s,t) \in E_{\text{async}}(f) \text{ and } \sigma = \text{proj}^I(t) \neq \text{proj}^I(s)\} & \text{if } \sigma \neq \epsilon, \\ \{t \in F | (s,t) \in E_{\text{async}}(f) \text{ and } \text{proj}^I(t) = \text{proj}^I(s)\} & \text{otherwise,} \end{cases}$$

$$\triangle(s,\sigma) = \begin{cases} A(s,\sigma) & \text{if } A(s,\sigma) \neq \emptyset, \\ \{q_{\text{failed}}\} & \text{otherwise.} \end{cases}$$

By construction the NFA-ϵ $\text{Auto}(f, I)$ accepts only words with no subsequent equal letters, since transitions between states which are projected to the same state by proj^I carry the weight ϵ. Furthermore, except for the initial state q_0 and q_{failed} all states of $\text{Auto}(f, I)$ are final states. The state q_{failed} has no outgoing transitions to other states. We give a small example, omitting the state q_{failed}, how such an automaton can look like:



(A) Auto$(f, \{1, 2\})$ omitting the state q_{failed} and transitions leading to it.

(B) Auto$(f, \{1\})$ omitting the state q_{failed} and transitions leading to it.

FIGURE 5.2.2. Two NFAs derived from Example 5.10.

Example 5.10. Consider an ASTG given by $f : \{0,1\}^2 \to \{0,1\}^2$, $(x_1, x_2) \mapsto (x_2, x_1)$. The two NFAs corresponding to the projection on the first – Auto$(f, \{1\})$ – and on both components – Auto$(f, \{1, 2\})$ – are depicted in Figure 5.2.2. We see that Auto$(f, \{1, 2\})$ essentially is the ASTG $G_{\text{async}}(f)$ if we omit the initial state q_0 and q_{failed} from the NFA-ϵ and add to each transition the destination state as a weight. In the case of Auto$(f, \{1, 2\})$ each word accepted by this NFA-ϵ corresponds to a walk in $G_{\text{async}}(f)$. If we consider the graph-representation of Auto$(f, \{1\})$ we see transitions with the weight ϵ - i.e. the transitions of the ASTG, where the first component does not change. If the first component changes, the new value of this component, is the weight of the corresponding transition. A word accepted by Auto$(f, \{1\})$ here, can correspond to several potential walks in $G_{\text{async}}(f)$. For example, the word 0 could stem from the sequence of states $q_0, 01$ or $q_0, 01, 00$ or simply $q_0, 00$.

To get some intuition for these sort of NFAs we give in the following some simple observations:

Remark 5.2. 1) For no $f : \{0,1\}^n \to \{0,1\}^n$ and $\emptyset \neq I \subseteq [n]$ the empty word is accepted by Auto(f, I).
2) Let w be any word of length bigger or equal 1. If a word $w\sigma$, $\sigma \in \Sigma$ is accepted by Auto(f, I) then w is accepted by Auto(f, I). Similar, if σw is accepted by Auto(f, I) then w is accepted by Auto(f, I).
3) Let $w = \sigma_1 \dots \sigma_k$, $k \in \mathbb{N}, k > 1$ a word over Σ accepted by Auto(f, I). Then for all $i \in [k-1]$ it holds $\sigma_i \neq \sigma_{i+1}$.

The first observation in Remark 5.2 simply follows from the fact that the initial state is no final state in Auto(f, I). The second one is correct since there is an edge to every state from the initial state q_0 of Auto(f, I).

Therefore, every sequence of states accepted by $\mathrm{Auto}(f, I)$ can be shortened from the left side remaining an accepted sequence. Similar arguments show that accepted words can be shortened from the right. The third observation follows since edges between states which are projected to the same state carry the weight ϵ.

A simple consequence of the second observation in Remark 5.2 is the following:

Remark 5.3. Assume w^1, w^2, w^3 are words over the alphabet $\{0,1\}^k$, $k \in [n]$ and w^2 has length ≥ 1. If the word $w^1w^2w^3$ is accepted by the NFA-ϵ $\mathrm{Auto}(f, I)$ then w^2 is accepted by $\mathrm{Auto}(f, I)$.

Proof. This is a simple consequence of applying the previous remark several time to the word $w^1w^2w^3$. □

We can also prove the following:

Proposition 5.8. The automaton $\mathrm{Auto}(f, I)$ accepts a word w corresponding to a walk $\tilde{p}$ in $\{0,1\}^{|I|}$ if and only if there is a walk p in $G_{\mathrm{async}}(f)$ such that $\mathrm{proj}^I(p) = \tilde{p}$.

Proof. " $\Leftarrow$ ": Assume there is a walk p in $G_{\mathrm{async}}(f)$ such that $\mathrm{proj}^I(p) = \tilde{p}$. Let us say $p = (s^1, \dots, s^k)$ for $k \in \mathbb{N}$. We want to show that $q_0, s^1, \dots, s^k$ is an accepting sequence. Then due to the definition of the NFA-ϵ $\mathrm{Auto}(f, I)$ for $i \in [k-1]$ $s^i \to s^{i+1}$ is a transition with weight

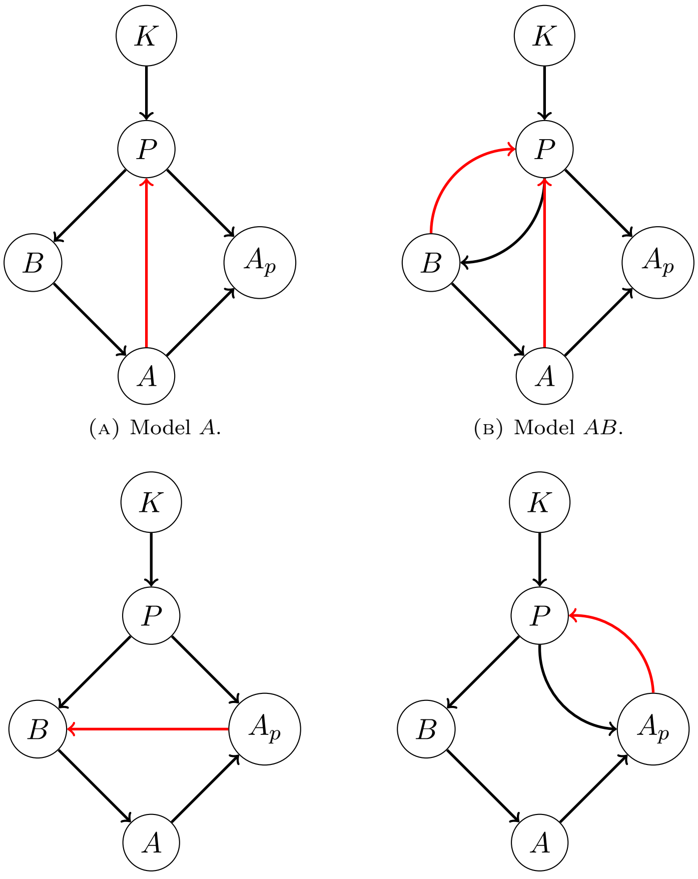
$$\begin{cases} \mathrm{proj}^I(s^{i+1}) & \text{if } \mathrm{proj}^I(s^{i+1}) \neq \mathrm{proj}^I(s^i), \\ \epsilon & \text{otherwise.} \end{cases}$$

In other words the sequence of projected states $\mathrm{proj}^I(s^1), \dots, \mathrm{proj}^I(s^k)$ after deleting subsequent equal elements – i.e. $\mathrm{proj}^I(p)$ – corresponds to the sequence of weights on the accepting sequence, which proves that the word $\tilde{p}$ is accepted by $\mathrm{Auto}(f, I)$.
" $\Rightarrow$ ": Any accepting sequence of states $q_0, s^1, \dots, s^k$ corresponds to a walk p from $s^1 \to \dots \to s^k$ in $G_{\mathrm{async}}(f)$. When projecting the walk p onto the components I, we obtain the words $\tilde{p}$, since transitions in $\mathrm{Auto}(f, I)$ carry only a weight different than ϵ if two subsequent states on the sequence of states p are different. □

Using the above proposition we are now ready to restate Problem 5.4.

Problem 5.5. For two Boolean functions f, g from $\{0,1\}^n \to \{0,1\}^n$ and $\emptyset \neq I \subseteq [n]$, which words w over the alphabets of $\mathrm{Auto}(f, I)$ and $\mathrm{Auto}(g, I)$ are only accepted by one of the two NFAs?

To solve the above problem we can use standard algorithms to obtain the NFA-ϵ accepting the symmetric difference of two NFAs, i.e. the language of words which are only accepted by one of the two NFAs but not by both Nozaki [1979]. If two automata $\mathrm{Auto}(f, I)$ and $\mathrm{Auto}(g, I)$ for $f, g \in \mathbb{B}(n, n)$ are equivalent for a set $\emptyset \neq I \subseteq [n]$ then we do not have a chance to distinguish the ASTGs $G_{\mathrm{async}}(f)$ and $G_{\mathrm{async}}(g)$ based on the components in I, since there is no sequence of states that can only stem from one of the two automata. The running time for checking for the equivalence of two



(A) Model A.

(B) Model AB.

(C) Model B.

(D) Model P.

FIGURE 5.2.3. Interaction graphs of four alternative Cytokinin-models Topcu-Alici [2012].

NFAs with state numbers m and n is bounded by $O(2^m + 2^n)$ Nozaki [1979]. Therefore, the size of the compared BNs must remain relatively small. If this is not the case one has to try to choose promising subnetworks.

5.2.2. Application to four models of a Cytokinin signaling pathway. We demonstrate the methods presented here on a small example. We consider four interaction graphs depicted in Figure 5.2.3. These four interaction graph represent four possible models of a Cytokinin signaling pathway suggested in Topcu-Alici [2012]. We use the results of Chapter 4 and consider the skeletons of these interaction graphs. More precisely, we consider the following Boolean functions with parameter $K \in \{0, 1\}$ representing the skeletons (see Section 4.3) of four interaction graphs stemming from the models A, AB, B, P (see Figure 5.2.3):

$$\begin{aligned}
f^A(P,B,A,A_p) &= \begin{pmatrix} P \oplus [(P \oplus^{-1} A) \vee (P \oplus K)] \\ B \oplus B \oplus P \\ A \oplus A \oplus B \\ A_p \oplus [(A_p \oplus A) \vee (A_p \oplus P)] \end{pmatrix} \\
&= \begin{pmatrix} 1 \oplus AK \oplus AP \oplus KP \oplus A \oplus P \\ P \\ B \\ AP \oplus AA_p \oplus PA_p \oplus A \oplus P \end{pmatrix} \\
&= \begin{pmatrix} \neg A \vee (B \wedge \neg P) \vee (\neg B \wedge P) \\ P \\ B \\ (A \wedge P) \vee (A \wedge \neg A_p) \vee (P \wedge \neg A_p) \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
f^{AB}(P,B,A,A_p) &= \begin{pmatrix} P \oplus [(P \oplus^{-1} A) \vee (P \oplus^{-1} B) \vee (P \oplus K)] \\ B \oplus B \oplus P \\ A \oplus A \oplus B \\ A_p \oplus [(A_p \oplus A) \vee (A_p \oplus P)] \end{pmatrix} \\
&= \begin{pmatrix} 1 \oplus AB \oplus KP \oplus ABK \oplus ABP \oplus AKP \oplus BKP \oplus P \\ P \\ B \\ AP \oplus AA_p \oplus PA_p \oplus A \oplus P \end{pmatrix} \\
&= \begin{pmatrix} (\neg A \wedge \neg B) \vee (\neg A \wedge \neg P) \vee (\neg B \wedge \neg P) \\ P \\ B \\ (A \wedge P) \vee (A \wedge \neg A_p) \vee (P \wedge \neg A_p) \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
f^{B}(P,B,A,A_p) &= \begin{pmatrix} P \oplus P \oplus K \\ B \oplus [(B \oplus P) \vee (B \oplus^{-1} A_p)] \\ A \oplus A \oplus B \\ A_p \oplus [(A_p \oplus P) \vee (A_p \oplus A)] \end{pmatrix} \\
&= \begin{pmatrix} K \\ 1 \oplus BP \oplus BA_p \oplus PA_p \oplus B \oplus A_p \\ B \\ AP \oplus AA_p \oplus PZ \oplus A \oplus P \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
f^{P}(P,B,A,A_p) &= \begin{pmatrix} P \oplus [(P \oplus K) \vee (P \oplus^{-1} A_p)] \\ B \oplus B \oplus P \\ A \oplus A \oplus B \\ A_p \oplus [(A_p \oplus P) \vee (A_p \oplus A)] \end{pmatrix} \\
&= \begin{pmatrix} 1 \oplus KP \oplus KA_p \oplus PA_p \oplus P \oplus A_p \\ P \\ B \\ AP \oplus AA_p \oplus PA_p \oplus A \oplus P \end{pmatrix}
\end{aligned}$$

The skeletons of model A and model B are depicted in Figure 5.2.4 and Figure 5.2.5.

Below we give some examples of components we could measure and corresponding sequences. For this purpose let us fix the value of K to 1.

- To distinguish $G_{\text{async}}(f^{AB})$ and $G_{\text{async}}(f^{A})$ we could measure:

  $\underline{B \text{ and } A}$: The sequence $00, 10, 00$ would indicate that f^A is not a realistic model, while f^{AB} would be possible. If we measure the

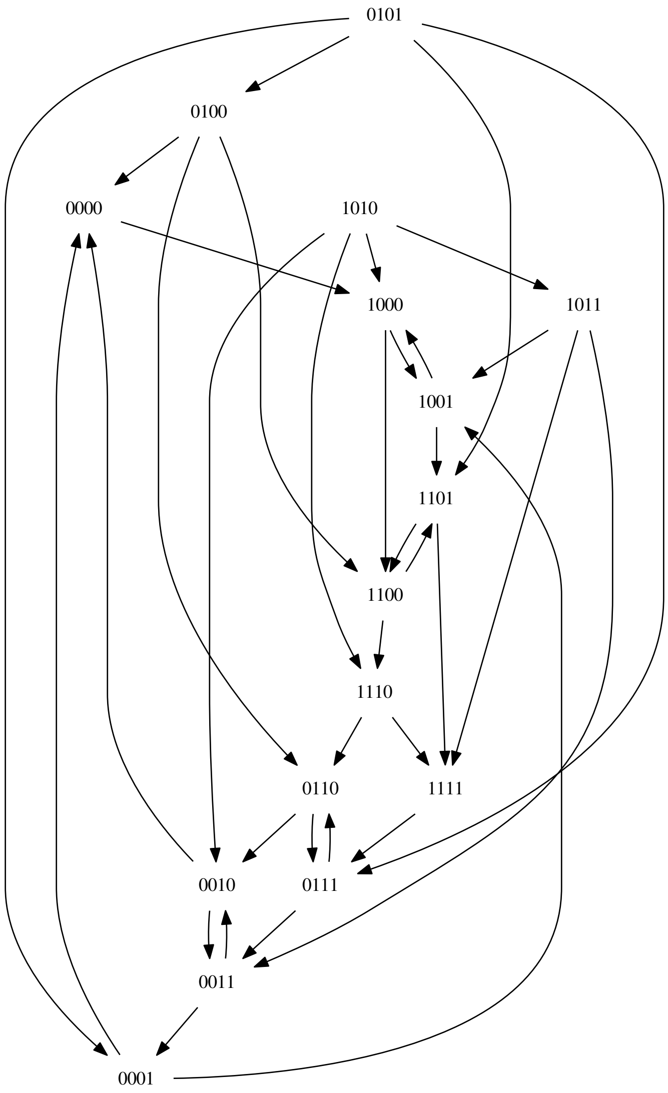


Figure 5.2.4. Model A.

components B and A in only two time points we do not have any chance to distinguish the two models.

- To distinguish $G_{\text{async}}(f^{AB})$ and $G_{\text{async}}(f^{B})$ we could measure:
  $\underline{P}$: The sequence $1 \rightarrow 0$ would indicate that f^B is not a realistic model, while f^{AB} would be possible.

  $\underline{P \text{ and } A}$: If we measure P and A in two points the sequences $10 \rightarrow 00$ or $11 \rightarrow 01$, corresponding to the previous case would indicate that f^B is not representing a realistic model, while f^{AB} could still explain this observation. In other words, it does not help to

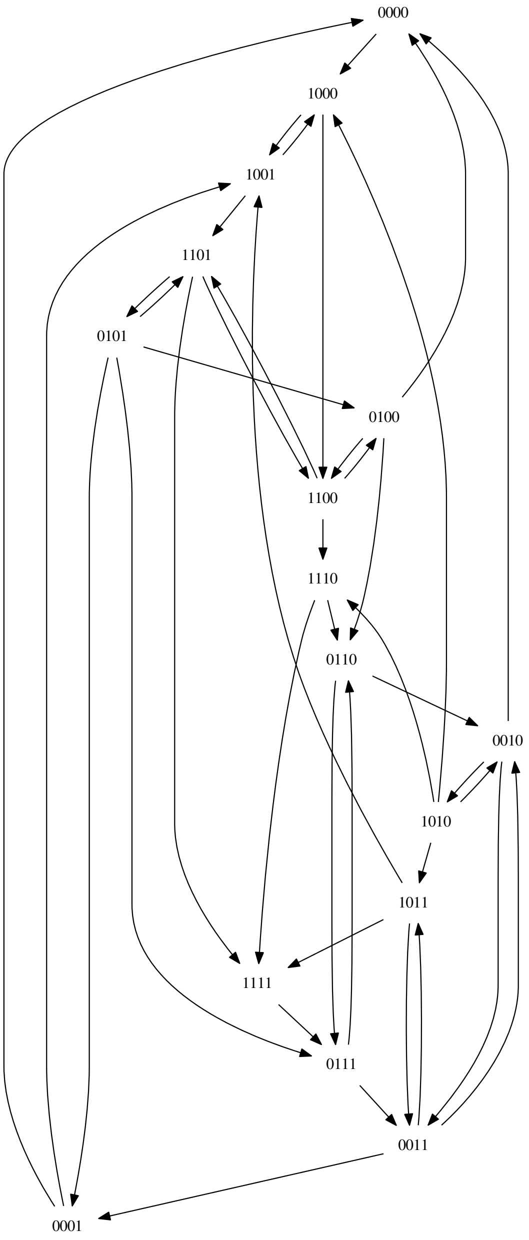


FIGURE 5.2.5. Model AB.

measure additionally the component A. However, if we would measure three instead of two time-points, we have a chance to observe time series that would speak against model f^{AB} (marked in red)

	f^{AB}	f^{B}
00,10,00	true	false
01,00,01	false	true
01,11,01	true	false
10,00,01	true	false
10,00,10	true	false
10,11,01	true	false
10,11,10	false	true
11,01,00	true	false
11,01,11	true	false
11,10,00	true	false

.

The remaining time series would be covered by measuring only P again.

- To distinguish $G_{\text{async}}(f^{AB})$ and $G_{\text{async}}(f^{P})$ we could measure:
  $\underline{B \text{ and } A_p}$: We need to measure at least three time-points and only the sequence $00 \to 10 \to 00$ could eliminate f^{P}.

5.3. Discussion

In this chapter we presented several ideas how different BNs or different possible behaviors of a single BN can be distinguished based on a subset of components of the BN. In Section 5.1 we presented formalizations of such criteria and we were able to use QBF solvers (e.g. Janota [2017] here) to find solutions in smaller systems. The reformulation of the State-Discrimination-Problem in Proposition 5.1 suggests that there exists no algorithm for this type of problems that will scale well. Nevertheless, we were able to demonstrate the usefulness of this approach on a a cell-fate decision model with 11 components in Section 5.1.3 from Calzone et al. [2010]. This suggests that for many biological models it is still possible to obtain helpful results. Also note that the original cell-fate decision model constructed in Calzone et al. [2010] contained 28 components. To analyze this model the original model was reduced in Calzone et al. [2010] using a network reduction method suggested in Naldi et al. [2011]. This shows that combining clever ways of reducing the Boolean network before using the methods presented here can enable us to tackle bigger systems. In Section 5.1.4 we gave another suggestion by reformulating the State-Discrimination-Problem into the language of algebraic geometry. We showed that the usage of Gröbner bases constitutes another possibility to compute the solutions. In the Boolean case there are relatively efficient computational tools for such tasks Brickenstein and Dreyer [2009]. Therefore, this could be a further promising approach to obtain better running times. Another problem is that the criteria in this chapter can lead to too many components that are necessary to measure. Therefore, it could be useful to relax the criteria in Section 5.1 by including probabilistic methods. I.e. allowing that the projected sets A and $B\backslash A$ in the State-Discrimination-Problem intersect to a certain degree.

Also the opposite approach – to analyze what can be said about a hypothesis given a fixed set of components we measure – is of interest.

This approach we followed in Section 5.2. Here, we used finite automata to analyze walks projected on some components to distinguish models. As a showcase we considered four interaction graphs. We computed the skeleton of these interaction graphs (see Section 4.3) and gave for a given set of components corresponding sequences of signs we need to measure to distinguish them. Also here we do not expect the methods presented to scale well with the number of components in BNs due to the use of nondeterministic finite automata. Methods for decreasing the average running time or suitable heuristic approaches should be investigated consequently here as well. Restricting the length of potential sequences a priori for example would be an easy approach to tackle bigger systems. Another problem of this approach is that measurements are not sufficient to observe a complete projected walk. There are also several theoretical aspects worth investigating still in the future. For example it would be interesting to investigate which ASTGs are equivalent with respect to certain output-nodes. Such ideas could be useful with respect to so-called network motifs, where the relation of certain reoccurring patterns in interaction graphs with the dynamics of corresponding BNs is suggested.

CHAPTER 6

Conclusion

In this thesis, several approaches to link ODE models and Boolean models of gene regulatory networks were considered. This is not solely of theoretical interest, but also of practical relevance as properties preserved across different classes of models of GRNs are unlikely to be model artifacts. This holds especially for Boolean models due to their simplicity. In light of this reasoning, the comparison between different modeling formalisms of GRNs can serve as an indicator as to the validity of results. At the beginning of this thesis, we set ourselves the task of determining the relations between these different classes of models. We made two contributions in this respect.

In Chapter 2 we studied a well-known algorithm that takes a Boolean function f as an input and outputs a family of ODE systems. We saw that the resulting ODE-system has trajectories resembling the discrete trajectories, but that in many cases there is no reasonable way to link the dynamics of the ODE-system with the dynamics of the corresponding Boolean network. However, looking at simple characteristics such as steady states or trap spaces, made it possible to prove that these properties are preserved during the conversion.

Unfortunately, this result alone is less than satisfying, given the fact, that a very large amount of analysis and research in Boolean networks is invested into properties of the state transition graphs relying on reachability properties. Conclusions based on such properties are thus built on a very shaky foundation. Firstly, the interpretation of Boolean states in the continuous setting is not straightforward. And secondly, even in those specific situations where the interpretation of continuous data is simpler, the "trajectories" found in Boolean models do not necessarily correspond to trajectories in the continuous models. Thus, we believe the construction of Boolean models should be more thoroughly linked to the behavior of ODEs.

In Chapter 4 we presented such an approach. There, we compared specific sets of ODE models with sets of Boolean models. For this purpose we considered the theory of qualitative differential equations. This constitutes a more general perspective on the relation between ODE systems and Boolean networks. Instead of looking at the concentrations of species in regulatory networks, we considered their dynamic trends, i.e. whether their concentrations are increasing or decreasing. In this approach, nothing more than the interaction graph of a regulatory network is necessary. To link these trends to Boolean networks, we reformulated a condition for possible sign changes in terms of a Boolean state transition graph. Afterwards, we showed that this Boolean state transition graph can be treated as the asynchronous state transition graph of a Boolean network. Thus, this approach constitutes an illustration of how Boolean networks can arise from families of ODEs, allowing an exact interpretation of states and transitions in the corresponding ODEs.

Also in Chapter 4, we were able to use a similar technique to analyze sets of Boolean models. We were able to obtain in this case an analogous Boolean state transition graph, which is essentially the same as the one obtained from sets of continuous models. This result is in and of itself interesting. Furthermore, such a graph can be effectively generated and reduced to the asynchronous state transition graph of a Boolean network. In principle, it can be used to investigate restrictions on the dynamics in certain sets of asynchronous Boolean networks. This is an area ripe for future investigation.

The focus of this work lies on the rather theoretical question of how different modeling formalisms of GRNs compare to one another. More generally, the overall goal, this thesis seeks to realize, is a good understanding of different modeling formalisms of GRNs and the ability to combine them efficiently without relying purely on heuristics. Despite this theoretical approach, we gave throughout this thesis ideas for applications of the presented results. We believe these ideas deserve more attention in the future and constitute a natural continuation of this research. In our opinion, the present goal should be to exploit the results in this thesis regarding the relation between different modeling frameworks for concrete applications.

The results in Chapter 2 on trap spaces open the door for combining Boolean and continuous models in better ways. Trap spaces could be used to facilitate parameter estimation by reducing the dynamics of an ODE model to trap spaces obtained from a coarser Boolean counterpart. This should be tested systematically for more models.

In Chapter 4, we considered a specific way of abstracting the dynamics of continuous models. However, this constitutes only a special case in the more general theory of qualitative reasoning Kuipers [1994]. In our opinion, it could be interesting to investigate different ways of abstracting ODEs and linking them to Boolean networks or, more generally, discrete multivalued networks.

Finally, the reduction of the Boolean state transition graphs to asynchronous Boolean networks considered in Chapter 4 constitutes a link between ODE models, interaction graphs and Boolean networks which deserves more attention. Similar to the results in Chapter 2, this yields the possibility of using Boolean networks to analyze models in different model classes. This was also the motivation to deviate in Chapter 5 from the original question of this thesis. Motivated by the relationship between Boolean networks, ODE models and interaction graphs, we began in Chapter 5 to explore ways to take advantage of these relationships. It will be interesting to see how such results will unfold in combination with the ideas of Chapter 5 in the future.

.

Bibliography

Wassim Abou-Jaoudé, Pauline Traynard, Pedro T Monteiro, Julio Saez-Rodriguez, Tomáš Helikar, Denis Thieffry, and Claudine Chaouiya. Logical modeling and dynamical analysis of cellular networks. *Frontiers in genetics*, 7:94, 2016.

Eva Ackermann, Eva Marie Weiel, Torsten Pfaff, and Barbara Drossel. Boolean versus continuous dynamics in modules with two feedback loops. *The European Physical Journal E*, 35(10):107, 1–13, 2012. ISSN 1292-8941. URL http://tubiblio.ulb.tu-darmstadt.de/59487/.

Bree B Aldridge, John M Burke, Douglas A Lauffenburger, and Peter K Sorger. Physicochemical modelling of cell signalling pathways. *Nature cell biology*, 8(11):1195, 2006.

Uri Alon. Design principles of biological circuits. *Febs J*, 277, 2007.

Jean-Pierre Aubin. *Viability Theory*. Birkhauser Boston Inc., Cambridge, MA, USA, 1991. ISBN 0-8176-3571-8.

Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008.

Naama Barkai and Stan Leibler. Robustness in simple biochemical networks. *Nature*, 387(6636):913, 1997.

Shohag Barman and Yung-Keun Kwon. A novel mutual information-based Boolean network inference method from time-series gene expression data. *PloS one*, 12(2):e0171097, 2017.

Grégory Batt, Delphine Ropers, Hidde de Jong, Johannes Geiselmann, Radu Mateescu, Michel Page, and Dominique Schneider. Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in escherichia coli. *Bioinformatics*, 21: 19–28, 2005.

Katinka Becker, Hannes Klarner, Melania Nowicka, and Heike Siebert. Designing mirna-based synthetic cell classifier circuits using Answer Set Programming. *Frontiers in bioengineering and biotechnology*, 6, 2018.

Marco Benedetti. skizzo: a suite to evaluate and certify qbfs. In *International Conference on Automated Deduction*, pages 369–376. Springer, 2005.

Béla Bollobás. *Modern graph theory*, volume 184. Springer Science & Business Media, 2013.

Michael Brickenstein and Alexander Dreyer. Polybori: A framework for gröbner-basis computations with Boolean polynomials. *Journal of Symbolic Computation*, 44(9):1326–1345, 2009.

Laurence Calzone, Laurent Tournier, Simon Fourquet, Denis Thieffry, Boris Zhivotovsky, Emmanuel Barillot, and Andrei Zinovyev. Mathematical modelling of cell-fate decision in response to death receptor engagement. *PLOS Computational Biology*, 6(3):1–15, 03 2010. doi: 10.1371/journal.

pcbi.1000702. URL https://doi.org/10.1371/journal.pcbi.1000702.

Guoliang Chai and Joseph G Gleeson. A newly discovered mechanism driving neuronal mutations in alzheimer's disease. *Nature*, 563(7733):631, 2018.

Madalena Chaves and Miguel Preto. Hierarchy of models: From qualitative to quantitative analysis of circadian rhythms in Cyanobacteria. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23(2):025113, 2013. doi: 10.1063/1.4810922. URL https://doi.org/10.1063/1.4810922.

Daizhan Cheng and Hongsheng Qi. Controllability and observability of boolean control networks. *Automatica*, 45(7):1659–1667, 2009.

David Cox, John Little, and Donal O'shea. *Ideals, varieties, and algorithms*, volume 3. Springer, 2007.

Yves Crama and Peter I. Hammer. *Boolean functions: Theory, algorithms, and applications.* 2011.

Hidde De Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103, 2004.

Peter Deuflhard and Susanna Röblitz. *A guide to numerical modelling in systems biology*, volume 12. Springer, 2015.

Cathérine Dupont, D Randall Armant, and Carol A Brenner. Epigenetics: definition, mechanisms and clinical perspective. In *Seminars in reproductive medicine*, volume 27, page 351. NIH Public Access, 2009.

Roderick Edwards, Hava Siegelmann, K Aziza, and Leon Glass. Symbolic dynamics and computation in model gene networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 11(1):160–169, 2001.

Klaus Eisenack. *Model ensembles for natural resource management: Extensions of qualitative differential equations using graph theory and viability theory.* PhD thesis, 2006.

Ettore Fornasini, Maria Elena Valcher, et al. Observability, reconstructibility and state observers of boolean control networks. *IEEE Trans. Automat. Contr.*, 58(6):1390–1401, 2013.

Sébastien Fuey, Pedro T Monteiro, Aurélien Naldi, Julien Dorier, Élisabeth Remy, and Claudine Chaouiya. Reversed dynamics to uncover basins of attraction of asynchronous logical models. In *Poster and oral presentation at ISMB/ECCB*, 2017.

Maximilien Gadouleau and Adrien Richard. On fixable families of boolean networks. *arXiv preprint arXiv:1804.01931*, 2018.

Sicun Gao, André Platzer, and Edmund M. Clarke. Quantifier elimination over finite fields using Gröbner bases. In Franz Winkler, editor, *Algebraic Informatics*, pages 140–157, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-21493-6.

Abhishek Garg, Alessandro Di Cara, Ioannis Xenarios, Luis Mendoza, and Giovanni De Micheli. Synchronous versus asynchronous modeling of gene regulatory networks. *Bioinformatics*, 24(17):1917–1925, 2008. doi: 10.1093/bioinformatics/btn336. URL +http://dx.doi.org/10.1093/bioinformatics/btn336.

Eva Gehrmann and Barbara Drossel. Boolean versus continuous dynamics on simple two-gene modules. *Physical Review E*, 82(4):046120, 2010.

Leon Glass and Stuart A. Kauffman. The logical analysis of continuous, nonlinear biochemical control networks. *Journal of theoretical biology*, 39(1):

103–129, April 1973. ISSN 0022-5193. doi: 10.1016/0022-5193(73)90208-7. URL https://doi.org/10.1016/0022-5193(73)90208-7.

A. Gonzalez Gonzalez, Aurélien Naldi, L. Sánchez, Denis Thieffry, and Claudine Chaouiya. GINsim: A software suite for the qualitative modelling, simulation and analysis of regulatory networks. *Biosystems*, 84(2):91 – 100, 2006. ISSN 0303-2647. doi: https://doi.org/10.1016/j.biosystems.2005.10.003. URL http://www.sciencedirect.com/science/article/pii/S0303264705001693. Dynamical Modeling of Biological Regulatory Networks.

Sylvain Goutelle, Michel Maurin, Florent Rougier, Xavier Barbaut, Laurent Bourguignon, Michel Ducher, and Pascal Maire. The hill equation: a review of its capabilities in pharmacological modelling. *Fundamental & clinical pharmacology*, 22(6):633–648, 2008.

Jean-Luc Gouzé and Tewfik Sari. A class of piecewise linear differential equations arising in biological models. *Dynamical systems*, 17(4):299–316, 2002.

John S. Griffith. Mathematics of cellular control processes i. negative feedback to one gene. *Journal of theoretical biology*, 20(2):202–208, 1968a.

John S. Griffith. Mathematics of cellular control processes ii. positive feedback to one gene. *Journal of Theoretical Biology*, 20(2):209–216, 1968b.

Geňa Hahn and Claude Tardif. Graph homomorphisms: structure and symmetry. In *Graph symmetry*, pages 107–166. Springer, 1997.

Inman Harvey and Terry Bossomaier. Time out of joint: Attractors in asynchronous random boolean networks. In *Proceedings of the Fourth European Conference on Artificial Life*, pages 67–75. MIT Press, Cambridge, 1997.

Archibald Vivian Hill. The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves. *j. physiol.*, 40:4–7, 1910.

Martin Hopfensitz, Christoph Müssel, Markus Maucher, and Hans A Kestler. Attractors in boolean networks: a tutorial. *Computational Statistics*, 1 (28):19–36, 2013.

Zoltán Horváth, Yunfei Song, and Tamás Terlaky. A novel unified approach to invariance in control. *arXiv preprint arXiv:1405.5167*, 2014.

Trey Ideker, Timothy Galitski, and Leroy Hood. A new approach to decoding life: systems biology. *Annual review of genomics and human genetics*, 2 (1):343–372, 2001.

Mahdi Imani and Ulisses Braga-Neto. Optimal gene regulatory network inference using the Boolean Kalman filter and multiple model adaptive estimation. In *2015 49th Asilomar Conference on Signals, Systems and Computers*, pages 423–427, Nov 2015. doi: 10.1109/ACSSC.2015.7421162.

François Jacob and Jacques Monod. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of molecular biology*, 3(3):318–356, 1961.

Mikoláš Janota. Qfun: towards machine learning in qbf. *arXiv preprint arXiv:1710.02198*, 2017.

Guy Karlebach and Ron Shamir. Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9(10):770–780, 2008.

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

Marcelline Kaufman, Christophe Soulé, and René Thomas. A new necessary condition on interaction graphs for multistationarity. *Journal of Theoretical Biology*, 248(4):675–685, 2007.

Hannes Klarner. *Contributions to the Analysis of Qualitative Models of Regulatory Networks*. PhD thesis, Freie Universität Berlin, 2015.

Hannes Klarner and Heike Siebert. Approximating attractors of Boolean networks by iterative CTL model checking. *Frontiers in Bioengineering and Biotechnology*, 3(130), 2015.

Hannes Klarner, Adam Streck, David Šafránek, Juraj Kolčák, and Heike Siebert. Parameter identification and model ranking of thomas networks. In *Computational Methods in Systems Biology*, pages 207–226. Springer, 2012.

Hannes Klarner, Alexander Bockmayr, and Heike Siebert. Computing maximal and minimal trap spaces of Boolean networks. *Natural Computing*, 14(4):535–544, 2015. ISSN 1567-7818. doi: 10.1007/s11047-015-9520-7. URL http://dx.doi.org/10.1007/s11047-015-9520-7.

Hannes Klarner, Adam Streck, and Heike Siebert. PyBoolNet-a python package for the generation, analysis and visualisation of Boolean networks. *Bioinformatics*, page btw682, 2016.

Hannes Klarner, Frederike Heinitz, Sarah Nee, and Heike Siebert. Basins of attraction, commitment sets and phenotypes of Boolean networks. *arXiv preprint arXiv:1807.10103*, 2018.

Edda Klipp, Ralf Herwig, Axel Kowald, Christoph Wierling, and Hans Lehrach. *Systems biology in practice: concepts, implementation and application*. John Wiley & Sons, 2008.

Jan Krumsiek, Sebastian Polsterl, Dominik M Wittmann, and Fabian J Theis. Odefy - From discrete to continuous models. *BMC Bioinformatics*, 11(233), 2010. URL https://doi.org/10.1186/1471-2105-11-233.

Benjamin Kuipers. Commonsense reasoning about causality: Deriving behavior from structure. *Artificial Intelligence*, 24(1-3):169–203, December 1984. doi: https://doi.org/10.1016/0004-3702(84)90039-0.

Benjamin Kuipers. *Qualitative reasoning: modeling and simulation with incomplete knowledge*. MIT press, 1994.

Harri Lähdesmäki, Ilya Shmulevich, and Olli Yli-Harja. On learning gene regulatory networks under the Boolean network model. *Machine Learning*, 52(1):147–167, Jul 2003. ISSN 1573-0565. doi: 10.1023/A:1023905711304. URL https://doi.org/10.1023/A:1023905711304.

Alexander Lange, Robert Schwieger, Julia Plöntzke, Stefan Schäfer, and Susanna Röblitz. Follicular competition in cows: the selection of dominant follicles as a synergistic effect. *Journal of mathematical biology*, pages 1–28, 2018.

Reinhard Laubenbacher and Brandilyn Stigler. A computational algebra approach to the reverse engineering of gene regulatory networks. *Journal of Theoretical Biology*, 229(4):523 – 537, 2004. ISSN 0022-5193. doi: https://doi.org/10.1016/j.jtbi.2004.04.037. URL http://www.sciencedirect.com/science/article/pii/S0022519304001754.

Tong Ihn Lee, Nicola J Rinaldi, François Robert, Duncan T Odom, Ziv Bar-Joseph, Georg K Gerber, Nancy M Hannett, Christopher T Harbison, Craig M Thompson, Itamar Simon, et al. Transcriptional regulatory networks in saccharomyces cerevisiae. *science*, 298(5594):799–804, 2002.

Julian Lewis. Autoinhibition with transcriptional delay: a simple mechanism for the zebrafish somitogenesis oscillator. *Current Biology*, 13(16):1398–1408, 2003.

Jinghang Liang and Jie Han. Stochastic boolean networks: an efficient approach to modeling gene regulatory networks. *BMC systems biology*, 6(1):113, 2012.

Shoudan Liang, Stefanie Fuhrman, and Roland Somogyi. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing*, 3:18–29, 1998.

Chao Luo and Xingyuan Wang. Dynamics of random boolean networks under fully asynchronous stochastic update based on linear representation. *PloS one*, 8(6):e66491, 2013.

Tarek Melliti, Damien Regnault, Adrien Richard, and Sylvain Sené. Asynchronous simulation of Boolean networks by monotone Boolean networks. In *International Conference on Cellular Automata*, pages 182–191. Springer, 2016.

Thomas Mestl, Erik Plahte, and Stig W Omholt. A mathematical framework for describing and analysing gene regulatory networks. *Journal of theoretical Biology*, 176(2):291–300, 1995.

Pejman Mohammadi, Niko Beerenwinkel, and Yaakov Benenson. Automated design of synthetic cell classifier circuits using a two-step optimization strategy. *Cell Systems*, 4(2):207 – 218.e14, 2017. ISSN 2405-4712. doi: https://doi.org/10.1016/j.cels.2017.01.003. URL http://www.sciencedirect.com/science/article/pii/S2405471217300030.

Pedro T. Monteiro, Wassim Abou-Jaoudé, Denis Thieffry, and Claudine Chaouiya. Model checking logical regulatory networks. *IFAC Proceedings Volumes*, 47(2):170 – 175, 2014. ISSN 1474-6670. doi: https://doi.org/10.3182/20140514-3-FR-4046.00135. URL http://www.sciencedirect.com/science/article/pii/S1474667015373985. 12th IFAC International Workshop on Discrete Event Systems (2014).

Mitio Nagumo. Über die Lage der Integralkurven gewöhnlicher Differentialgleichungen. *Nippon Sugaku-Buturigakkwai Kizi Dai 3 Ki*, 24(0):551–559, 1942.

Aurélien Naldi, Élisabeth Remy, Denis Thieffry, and Claudine Chaouiya. Dynamically consistent reduction of logical regulatory graphs. *Theoretical Computer Science*, 412(21):2207 – 2218, 2011. ISSN 0304-3975. doi: https://doi.org/10.1016/j.tcs.2010.10.021. URL http://www.sciencedirect.com/science/article/pii/S0304397510005839. Selected Papers from the 7th International Conference on Computational Methods in Systems Biology.

Mathilde Noual. *Updating Automata Networks. (Mises à jour de réseaux d'automates)*. PhD thesis, École normale supérieure de Lyon, France, 2012. URL https://tel.archives-ouvertes.fr/tel-00726560.

Akihiro Nozaki. Equivalence problem of non-deterministic finite automata. *Journal of Computer and System Sciences*, 18(1):8–17, 1979.

Ertuğrul M Özbudak and Julian Lewis. Notch signalling synchronizes the zebrafish segmentation clock but is not needed to create somite boundaries. *PLoS genetics*, 4(2):e15, 2008.

Remigijus Paulavičius and Julius Žilinskas. Analysis of different norms and corresponding Lipschitz constants for global optimization. *Technological and Economic Development of Economy*, 12(4):301–306, 2006.

Loïc Paulevé and Adrien Richard. Static analysis of boolean networks based on interaction graphs: A survey. *Electronic Notes in Theoretical Computer Science*, 284:93–104, 2012.

Wolf Reik. Stability and flexibility of epigenetic gene regulation in mammalian development. *Nature*, 447(7143):425, 2007.

Élisabeth Remy and Paul Ruet. From minimal signed circuits to the dynamics of Boolean regulatory networks. *Bioinformatics*, 24(16):i220–i226, 2008.

Élisabeth Remy, Paul Ruet, and Denis Thieffry. Graphic requirements for multistability and attractive cycles in a Boolean dynamical framework. *Advances in Applied Mathematics*, 41(3):335–350, 2008.

Adrien Richard and Jean-Paul Comet. Stable periodicity and negative circuits in differential systems. *Journal of Mathematical Biology*, 63(3): 593–600, 2010. ISSN 1432-1416. doi: 10.1007/s00285-010-0388-y. URL http://dx.doi.org/10.1007/s00285-010-0388-y.

Keijo Ruohonen. Graph theory, 2013. URL http://math.tut.fi/~ruohonen/GT_English.pdf.

Assieh Saadatpour and Réka Albert. A comparative study of qualitative and quantitative dynamic models of biological regulatory networks. *EPJ Nonlinear Biomedical Physics*, 4(1):1–13, 2016.

Julio Saez-Rodriguez, Luca Simeoni, Jonathan A Lindquist, Rebecca Hemenway, Ursula Bommhardt, Boerge Arndt, Utz-Uwe Haus, Robert Weismantel, Ernst D Gilles, Steffen Klamt, et al. A logical model provides insights into T cell receptor signaling. *PLoS computational biology*, 3(8):e163, 2007.

Regina Samaga, Julio Saez-Rodriguez, Leonidas G Alexopoulos, Peter K Sorger, and Steffen Klamt. The logic of egfr/erbb signaling: theoretical properties and analysis of high-throughput data. *PLoS computational biology*, 5(8):e1000438, 2009.

Moises Santillán. On the use of the hill functions in mathematical models of gene regulatory networks. *Mathematical Modelling of Natural Phenomena*, 3(2):85–97, 2008.

S Shankar Sastry. *Nonlinear systems: Analysis, stability, and control*, volume 10. Springer Science & Business Media, 2013.

Yosuke Sato, Shutaro Inoue, Akira Suzuki, Katsusuke Nabeshima, and Ko Sakai. Boolean gröbner bases. *Journal of symbolic computation*, 46 (5):622–632, 2011.

Rebekka Schlatter, Kathrin Schmich, Ima Avalos Vizcarra, Peter Scheurich, Thomas Sauter, Christoph Borner, Michael Ederer, Irmgard Merfort, and Oliver Sawodny. On/off and beyond-a boolean model of apoptosis. *PLoS computational biology*, 5(12):e1000595, 2009.

Rebekka Schlatter, Nicole Philippi, Gaby Wangorsch, Robert Pick, Oliver Sawodny, Christoph Borner, Jens Timmer, Michael Ederer, and Thomas Dandekar. Integration of boolean models exemplified on hepatocyte signal transduction. *Briefings in bioinformatics*, 13(3):365–376, 2011.

Robert Schwieger and Heike Siebert. Graph representations of monotonic Boolean model pools. *Computational Methods in Systems Biology. CMSB 2017. Lecture Notes in Computer Science*, 10545, 2017.

Robert Schwieger and Heike Siebert. Representing model ensembles as boolean functions. In Patrick Amar, François Képès, Vic Norris, and Attila Csikasz-Nagy, editors, *advances in Systems and Synthetic Biology*, pages 102–116, Evry, France, Mar 2018. ISBN 978-2-7598-2266-9.

Robert Schwieger, Heike Siebert, and Susanna Röblitz. Correspondence of trap spaces in different models of bioregulatory networks. *SIAM Journal on Applied Dynamical Systems*, 17(2):1742–1765, 2018.

Lee A Segel. On the validity of the steady state assumption of enzyme kinetics. *Bulletin of mathematical biology*, 50(6):579–593, 1988.

Ilya Shmulevich, Edward R Dougherty, and Wei Zhang. From boolean to probabilistic boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE*, 90(11):1778–1792, 2002.

Heike Siebert. Deriving behavior of Boolean bioregulatory networks from subnetwork dynamics. *Mathematics in Computer Science*, 2(3):421–442, Mar 2009. ISSN 1661-8289. doi: 10.1007/s11786-008-0064-4. URL https://doi.org/10.1007/s11786-008-0064-4.

El Houssine Snoussi. Qualitative dynamics of piecewise-linear differential equations: A discrete mapping approach. *Dynamics and Stability of Systems*, 4(3-4):565–583, 1989. doi: 10.1080/02681118908806072. URL http://dx.doi.org/10.1080/02681118908806072.

El Houssine Snoussi. Necessary conditions for multistationarity and stable periodicity. *Journal of Biological Systems*, 06(01):3–9, 1998. doi: 10.1142/S0218339098000042.

Christophe Soulé. Graphic requirements for multistationarity. *ComPlexUs*, 2003.

Claudia Stötzel, Susanna Röblitz, and Heike Siebert. Complementing ODE-based system analysis using Boolean networks derived from an Euler-like transformation. *PloS one*, 10(10):e0140954, 2015.

Adam Streck. *Toolkit for Reverse Engineering of Molecular Pathways via Parameter Identification*. PhD thesis, Freie Universität Berlin, 2015.

Adam Streck and Heike Siebert. Extensions for ltl model checking of thomas networks. *Advances is Systems and Synthetic Biology*, 14:101–114, 2015.

Adam Streck, Kirsten Thobe, and Heike Siebert. Data-driven optimizations for model checking of multi-valued regulatory networks. *Biosystems*, 149: 125–138, 2016.

Bernd Sturmfels. *Gröbner bases and convex polytopes*, volume 8. American Mathematical Soc., 1996.

Motoyosi Sugita. Functional analysis of chemical systems in vivo using a logical circuit equivalent. ii. the idea of a molecular automaton. *Journal of Theoretical Biology*, 4(2):179–192, 1963.

Zoltan Szallasi, Jörg Stelling, and Vipul Periwal. System modeling in cell biology: From concepts to nuts and bolts. 2006.

Gerald Teschl. *Ordinary differential equations and dynamical systems*, volume 140. American Mathematical Society Providence, 2012.

Sven Thiele, Sandra Heise, Wiebke Hessenkemper, Hannes Bongartz, Melissa Fensky, Fred Schaper, and Steffen Klamt. Designing optimal experiments to discriminate interaction graph models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2018.

René Thomas. Boolean formalization of genetic control circuits. *Journal of theoretical biology*, 42(3):563–585, 1973.

René Thomas. Logical analysis of systems comprising feedback loops. *Journal of Theoretical Biology*, 73(4):631–656, 1978.

René Thomas. On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations. In *Numerical methods in the study of critical phenomena*, pages 180–193. Springer, 1981.

René Thomas. Regulatory networks seen as asynchronous automata: A logical description. *Journal of Theoretical Biology*, 153(1):1–23, 1991.

René Thomas and Marcelle Kaufman. Multistationarity, the basis of cell differentiation and memory. i. structural conditions of multistationarity and other nontrivial behavior. *Chaos*, 11(1):170–179, 2001a. doi: http://dx.doi.org/10.1063/1.1350439. URL http://scitation.aip.org/content/aip/journal/chaos/11/1/10.1063/1.1350439.

René Thomas and Marcelle Kaufman. Multistationarity, the basis of cell differentiation and memory. ii. logical analysis of regulatory networks in terms of feedback circuits. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 11(1):180–195, 2001b.

René Thomas, Anne-Marie Gathoye, and Lucie Lambert. A complex control circuit: Regulation of immunity in temperate bacteriophages. *European Journal of Biochemistry*, 71(1):211–227, 1976.

Marcus J Tindall, Eamonn A Gaffney, Philip K Maini, and Judith P Armitage. Theoretical insights into bacterial chemotaxis. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 4(3):247–259, 2012.

Elisa Tonello, Etienne Farcot, and Claudine Chaouiya. Local negative circuits and cyclic attractors in boolean networks with at most five components. *SIAM Journal on Applied Dynamical Systems*, 18(1):68–79, 2019.

Firdevs Topcu-Alici. Modellierung und Analyse der Regulationsmechanismen des Cytokinin- Signalwegs in Arabidopsis thaliana. Master's thesis, Freie Universität Berlin, 2012.

John J Tyson. Modeling the cell division cycle: cdc2 and cyclin interactions. *Proceedings of the National Academy of Sciences*, 88(16):7328–7332, 1991.

Alan Veliz-Cuba, Joseph Arthur, Laura Hochstetler, Victoria Klomps, and Erikka Korpi. On the relationship of steady states of continuous and discrete models arising from biology. *Bulletin of Mathematical Biology*, 74 (12):2779–2792, 2012. ISSN 0092-8240. doi: 10.1007/s11538-012-9778-1. URL http://dx.doi.org/10.1007/s11538-012-9778-1.

Alan Veliz-Cuba, Boris Aguilar, Franziska Hinkelmann, and Reinhard Laubenbacher. Steady state analysis of Boolean molecular network models via model reduction and computational algebra. *BMC Bioinformatics*, 15 (1):221, Jun 2014a. ISSN 1471-2105. doi: 10.1186/1471-2105-15-221. URL https://doi.org/10.1186/1471-2105-15-221.

Alan Veliz-Cuba, Ajit Kumar, and Krešimir Josić. Piecewise linear and Boolean models of chemical reaction networks. *Bulletin of Mathematical Biology*, 76(12):2945–2984, Dec 2014b. ISSN 1522-9602. doi: 10.1007/s11538-014-0040-x. URL https://doi.org/10.1007/s11538-014-0040-x.

Paola Vera-Licona, Abdul Jarrah, Luis David Garcia-Puente, John McGee, and Reinhard Laubenbacher. An algebra-based method for inferring gene regulatory networks. *BMC systems biology*, 8(1):37, 2014.

Santiago Videla, Julio Saez-Rodriguez, Carito Guziolowski, and Anne Siegel. caspo: a toolbox for automated reasoning on the response of logical signaling networks families. *Bioinformatics*, page btw738, 2017.

Rui-Sheng Wang, Assieh Saadatpour, and Reka Albert. Boolean modeling in systems biology: an overview of methodology and applications. *Physical biology*, 9(5):055001, 2012.

Dominik M Wittmann, Jan Krumsiek, Julio Saez-Rodriguez, Douglas A Lauffenburger, Steffen Klamt, and Fabian J Theis. Transforming Boolean models to continuous models: Methodology and application to T-cell receptor signaling. *BMC Systems Biology*, 3(1):98, 2009. doi: 10.1186/1752-0509-3-98. URL http://dx.doi.org/10.1186/1752-0509-3-98.

Jorge G. T. Zañudo and Réka Albert. An effective network reduction approach to find the dynamical repertoire of discrete dynamic networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23(2):025111, 2013. doi: 10.1063/1.4809777. URL https://doi.org/10.1063/1.4809777.

Jorge G. T. Zañudo and Réka Albert. Cell fate reprogramming by control of intracellular network dynamics. *PLOS Computational Biology*, 11(4):1–24, 04 2015. doi: 10.1371/journal.pcbi.1004193. URL https://doi.org/10.1371/journal.pcbi.1004193.

Kuize Zhang and Lijun Zhang. Observability of boolean control networks: A unified approach based on finite automata. *IEEE Transactions on Automatic Control*, 61(9):2733–2738, 2016.

Lintao Zhang and Sharad Malik. Conflict driven learning in a quantified boolean satisfiability solver. In *Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design*, pages 442–449. ACM, 2002.

Index

abstraction, 65
activating edge, 21
alphabet, 119
ASTG, 18
ASTG of a projected ASTG, 53
asymptotically stable fixed point, 12
asynchronous Boolean network, 18
asynchronous state transition graph, 18
attractor, 9
attractors, 9

Boolean expression, 10
Boolean formula, 10
Boolean monotonic model pool, 73, 74
Boolean network, 17
Boolean Strong Nullstellensatz, 114

cardinality of a set, 8
circuit, 9
clause, 11
CNF, 11
complement of an ASTG, 46
components of a Boolean network, 17
conjunctive normal form, 11
continuous monotonic model pool, 60, 63
cooperative binding, 14
cycle, 9

degradation terms, 25
difference of ASTGs, 46
directed graph, 8
discrete derivative, 20
disjunctive normal form, 11
dissociation constant, 14
DNF, 11
dynamical system, 12

edges, 8
empty string, 119
epigenetics, 2
ϵ-extensions, 120
essential component, 103
evolution function, 12

field polynomial, 112
fixed point, 12
flow, 12

gene regulatory network(GRN), 1
global interaction graph, 20
Gröbner basis, 116
graph homomorphism, 10
graph interaction graph, 21
graph isomorphism, 10

Hamming distance, 11
Hill coefficient, 24
Hill cube, 29
Hill equation, 15
Hill function, 24
Hill functions, 15

I-subspace, 19
identity function, 45
indegree, 9
indicator function, 102
inhibitory edge, 21
initial ideal, 116
initial monomial, 116
interaction graph, 20
intersection of ASTGs, 46
invariant, 12

letter, 119
life-times, 25
ligand, 14
literal, 11
local interaction graph, 20
loop, 8

mass action kinetics, 13
Messenger RNA (mRNA), 1
minimal Gröbner basis, 116
monomial ordering, 115
monotonic function, 74
multivariate polynomial interpolation, 23

NFA-ϵ, 119
no-return set, 9
nodes, 8
nondeterministic finite automaton, 119
normalized Hill cube, 29

orbit through x, 12
outdegree, 9

path, 9
phase space, 12
predecessor, 8
projected walk, 119

projection, 19, 50

QDE, 60
QSTG, 60
qualitative differential equations, 60
qualitative models, 3
qualitative reasoning, 5
qualitative state transition graph, 60
qualitative state transition graph (QSTG), 65
qualitative states, 65
qualitative transitions, 65
quantitative models, 3
quotient graph, 48

reachable, 9
reasonable, 63
recall, 94
reduced Gröbner basis, 116
regulatory site, 1

skeleton of Σ, 89
skeleton of the qualitative state transition graphs, 61
space of admissible trajectories, 63
stable fixed point, 12
standard monomial, 116
state, 17, 48
state space, 12
state transition graph, 17
steady node, 9
steady state, 9
strong basin of attraction, 10
subspace, 19
successors, 8
symmetric difference of ASTGs, 46
Synchronous Boolean network, 17
synchronous Boolean network, 17
synchronous state transition graph, 18
Systems biology, 2

tangent cone, 39
term, 11
threshold, 24
trail, 9
trajectory through x, 12
transcription factor, 1
transition, 17, 48
transitive closure, 9
trap set, 9
trap space, 19, 31
trap space, 19

union of ASTGs, 46
update function, 17
update-rule, 17

vanishing ideal, 112

walk, 9
weak basin of attraction, 10
word, 119

Appendix

Zusammenfassung

Diese Arbeit beschäftigt sich mit unterschiedlichen Typen von qualitativen Modellen genregulatorischer Netzwerke. Vier Typen von Modellen werden betrachtet:

(1) Interaktionsgraphen
(2) Boolesche Netzwerke
(3) Modelle, die auf Differentialgleichungen basieren
(4) Diskrete Abstraktionen von Differentialgleichungen

Wir werden mehr über die Beziehungen zwischen diesen Modellgruppen lernen und wie diese Beziehungen genutzt werden können, um einzelne Modelle zu analysieren. Der Schwerpunkt liegt hierbei auf der mathematischen Analyse dieser Modellgruppen. In dieser Hinsicht leistet diese Arbeit mehrere Beiträge.

Zunächst betrachten wir Boolesche Netzwerke und parametrisierte Familien von gewöhnlichen Differentialgleichungen (ODEs). Um solche ODE-Modelle systematisch aus Booleschen Modellen abzuleiten, wurden in der Vergangenheit verschiedene automatische Konvertierungsalgorithmen vorgeschlagen. In Kapitel 2 werden einige dieser Algorithmen näher untersucht. Wir werden beweisen, dass bestimmte invariante Mengen bei der Konvertierung eines Booleschen Modells in ein ODE-Modell erhalten bleiben.

Der zweite Ansatz, der in dieser Arbeit verfolgt wird, beschäftigt sich mit diskreten Abstraktionen der Dynamik von Modellen. Mit Hilfe dieser Abstraktionen ist es möglich, die Struktur – den Interaktionsgraphen – und die Dynamik der zugehörigen Modelle in Bezug zu setzen. Diese Methode wird sowohl auf Boolesche Modelle als auch auf ODE-Modelle angewandt. Gleichzeitig erlaubt dieser Ansatz Mengen von Modellen in unterschiedlichen Modellgruppen zu vergleichen, die dieselbe Struktur haben.

Die abstra-hierten Dynamiken (genauer die Einschränkungen der abstrahierten Dynamiken) der Booleschen Modellmengen oder ODE-Modellmengen können als Boolesche Zustandsübergangsgraphen repräsentiert werden. Wir werden zeigen, dass diese Zustandsübergangsgraphen wiederum selber als (asynchrone) Boolesche Netzwerke aufgefasst werden können.

Trotz der theoretischen Ausgangsfrage werden in dieser Arbeit zahlreiche Anwendungen aufgezeigt. Die Ergebnisse aus Kapitel 2 können zur Modellreduktion benutzt werden, indem die Dynamik der ODE-Modelle auf den zu den Booleschen Netzwerken gehörigen “trap spaces” betrachtet wird. Die Resultate aus Kapitel 4 können zur Netzwerkinferenz oder zur Analyse von Modellmengen genutzt werden. Weiterhin werden im letzten Kapitel dieser Arbeit einige Anwendungsideen im Bezug auf Experimentdesign eingeführt.

Dies führt zu der Fragestellung, wie verschiedene asynchrone Boolesche Netzwerke oder unterschiedliche Dynamiken, die mit einem einzelnen Modell vereinbar sind, unterschieden werden können.

Declaration

I assert to have written this PhD thesis on my own, that I cited all the sources I used, and that I did not use any illegitimate tools.

Ich versichere diese Arbeit selbstständig, ohne unerlaubte Hilfsmittel verfasst und keine außer den angeführten Quellen verwendet zu haben.

Robert Schwieger
July 24, 2019