# Logic-based Modeling

## in Systems Biology

Dr.rer.nat. Heike Siebert

Habilitationsschrift
im Fach Mathematik

Freie Universität Berlin
Juni 2011

# Contents

# Introduction and Overview

Mathematics, by its universal nature, is nowadays intertwined with a wide variety of disciplines, such as physics, economics, social or life sciences. Phrasing a problem in rigorous mathematical terms often allows for a high-level understanding and opens the way for structured analysis. In turn, problems and questions found in applications have the potential to spark profound mathematical questions of interest, also on a purely theoretical level. This monograph focusses on logic-based modeling approaches in systems biology and some of the mathematical challenges we encounter in this area of application.

In systems biology, the goal is to understand how molecular components such as genes or proteins interact to give rise to specific behavior. The control of essential cellular processes, such as proliferation, differentiation and apoptosis, is driven by an intricate interplay of many different molecule species interacting through a variety of regulatory mechanisms. Large and complex interaction networks are thus delineated, calling for dedicated modelling methods to understand and predict their behaviors.

A natural and well-established first step in modeling is the use of graphs to capture network structure, i. e., components and dependencies between them. A wide range of characteristics of interest for biological systems can be efficiently analyzed using graph theoretical concepts. However, such models are intrinsically static and many important features arising from the dynamical rules governing the system are not recoverable [3]. In contrast, differential equation models are dynamic models that provide a high degree of precision and offer good representations of underlying biochemical rate laws. However, construction and validation are difficult for systems where quantitative data and knowledge of mechanistic details and kinetic parameters are scarce, as is often the case when modeling biological networks [27, 64].

Discrete or logic-based modeling represents a compromise between complex and highly specified differential equation models and a purely structural representation. Based on qualitative information, the model generates dynamics represented as time and state discrete transition systems. Albeit a rather coarse representation of a biological system, such models are capable of highlighting fundamental characteristics of network structure and behavior [19, 38, 11].

Already in the 1960s, S. Kauffman used discrete models in a systems biology context, namely for analyzing gene regulatory networks [34]. He chose a very abstract representation of such systems as Boolean networks, i.e., interpreting genes as Boolean variables and capturing the effects of interactions in Boolean functions. At the time, modeling of gene regulatory networks was severely hampered by lack of data. Often, the available information did not even suffice to extract a well-founded idea of the network structure. So, rather than modeling a specific system, Kauffman investigated the characteristics of random Boolean networks. First, he generated a random directed graph, called interaction or dependency graph, representing the network

structure. Here, the vertices of the graph correspond to the network components while edges signify dependencies between components. The network components are interpreted as Boolean variables, where the value 1 represents gene activity. The state of the network is then described by a Boolean vector of values for all network components.

The structural information encoded in the interaction graph is usually not enough to determine the network dynamics. The behavior of each component is described by a Boolean function on the state space which depends only on the values of the predecessors of that component in the interaction graph. However, the way the input from the predecessors is processed has to be specified, e.g., as an AND or OR gate. This amounts to deciding on the values of a truth table for each component. So, in essence, the network structure imposes some restrictions on the function by determining the set of components the function depends on, but leaves the specific choice of function open. To obtain a fully specified model, one function from the set of functions in agreement with the network structure has to be chosen, i.e., the values of the function have to be designated. This process is often referred to as specifying the (discrete or logical) parameters of the system.

Again, Kauffman chose the parameters randomly to specify the system. The dynamics is now derived from the Boolean function, often called update function. The system transitions from a given state to its image state. This method to derive state transitions is called synchronous update. Since the state space is finite, the dynamics of the system can also be represented by a finite directed graph, the state transition graph, where the vertices correspond to the states and edges lead from each state to its image.

Analysis of the system focusses both on the network structure and its dynamics. Many graph theoretical characteristics, such as connectivity or density measures, are of interest when analyzing the structure of a biological system, since they might indicate the suitability of a network for specific functions and allow insights into characteristics such as robustness and adaptability. The dynamics can be analyzed using iteration theory, trajectories are given by sequences of iterates and system attractors by fixed or periodic points of the update function. The terminology can easily be translated into graph theoretical notions applicable to the state transition graph, where trajectories are represented by paths and attractors by terminal strongly connected components.

Within this context, Kauffman and others were able to determine dynamical characteristics of ensembles of random Boolean networks sharing structural properties [35, 23, 36]. Although his research was clearly mathematical in nature and, at this point, rather far away from actual application, most qualitative modeling approaches today share the core of his method: representation of the system as a discrete function based on network structure and logical parameters.

Kauffman's ideas were extended by several authors to obtain formalisms more suited for modeling biological systems [61, 22, 53]. R. Thomas and colleagues, for example, made the step from Boolean to multi-valued variables to allow for a finer distinction of activity levels of network components, and introduced an update scheme for the function describing network dynamics yielding a more realistic representation of the system's behavior [62]. The so-called asynchronous update acknowledges the fact that the processes represented by a component's value change in the model can be vastly different in reality. In particular, it signifies a strong over-simplification to assume all processes involved in a state transition are executed in the same amount of time. The asynchronous update, in contrast, only allows update of one component at a time, i.e., a state and its successor only differ in one component value. Since no assumption is being made on which component will change its value first, in case of multiple components to

be updated, the formalism results in a non-deterministic representation of the dynamics. In the corresponding state transition graph, the out-degree of a vertex will thus be generally greater than one, reflecting the different choices of components that are updated. In addition, each component is only updated in unitary steps, resulting in a gradual evolution of activity level change mimicking, e.g., the continuous increase or decrease of concentration values. Although the asynchronous state transition graph is harder to analyze than its synchronous counterpart, the additional potential of this method to capture realistic behavior has been shown in a variety of applications [52, 1, 42].

Today, discrete formalisms are a well-established alternative or complement to quantitative modeling methods, since they can be utilized even if quantitative data and mechanistic knowledge is lacking. A wide range of discrete modeling techniques and tools is available, including, e. g., Petri net, automata, and algebraic formalisms [38, 30, 14]. However, the actual modeling process of a biological system remains an intricate task best tackled in interdisciplinary teams, resulting in the typical research cycle of systems biology [37].

Given a system of interest, the cycle starts off with an initial modeling process where the available information is translated into a mathematical description. In the case of qualitative modeling, this amounts to determining a set of network components to be modeled and a logic-based description of the interactions between them in order to come up with a specified model. This process is by no means trivial and the development of methods allowing to infer a suitable network model from a variety of biological data is an active research area. Often, the available information is not sufficient to derive a fully parameterized model. Biologically motivated hypotheses, for example, can then be used to obtain a specified model, whose analysis may yield further insight into the system.

As already mentioned, analysis of a regulatory network includes structural and dynamical aspects. The interaction graph is mostly well-accessible for graph theoretical analysis methods. When considering the dynamical behavior, we face the difficulty of having to analyze an exponentially larger complex graph. Providing efficient methods for a comprehensive analysis of the state transition graph is consequently an active research area. A particular focus of the analysis lies on uncovering essential features of the network as well as on determining network characteristics verifiable through feasible biological experiments.

Acquisition of new data tailored to answer questions about the existing model usually starts the process of model refinement. If the current model is not in agreement with the additional data, the model has to be adapted. This starts an iterative process of model building, model analysis and experimental work resulting in new data. During this process, we have to keep in mind that the modeling formalism should ideally fully exploit the available data and has to be suited to the research questions of interest. The qualitative approach is very abstract and might not be capable of reproducing behavior heavily depending on, e.g., small fluctuations of concentrations or stochastic effects. Model refinement may therefore also result in transferring the model into a different, more suited, formalism, raising the level of detail incorporated in the model in step with the experimental results.

The short description of the research cycle already hints at the wide range of interesting mathematical challenges inherent in interdisciplinary research in systems biology. The different contributions collected in this work touch upon problems related to different stages in the research cycle and are presented in three blocks corresponding to the tasks of model building, analysis and refinement. However, it will become clear that this classification, as the division of the research cycle, is not clear-cut. In the following, I will give an introduction to these

three topics to put my respective contributions into context. In addition, I provide a detailed summary of the results presented in this work.

### Model building

As mentioned, a logic-based model consists of an interaction graph representing the network structure and a set of logical parameters that specify the way components influence each other and give rise to dynamical behavior. Often, biologists already have a fair idea about the structure underlying a regulatory network, but parameters are not or only partially known. Due to the discrete nature of a logic-based model, the parameter space of a model pool corresponding to a given interaction graph is finite. So, in theory, all the models generated by the parameter space could be analyzed, but due to combinatorial explosion such an approach would only be feasible for very small systems. To constrain the set of possible parameters, the information available on the system has to be evaluated carefully for its suitability for parameter inference. The qualitative character of the approach allows to exploit a variety of biological data, ranging from quantitative time series to rough observations such as the system being able to display different stable behaviors.

In order to utilize available information for parameter inference, it has to be translated into characteristics that can be matched to properties of the state transition graph and the included state transitions. This means quantitative information, e.g., real-valued concentration values, has to be discretized in order to relate it to the states of the discrete model. Data discretization is a delicate issue, since the results directly influence the choice of model, which holds true for all data processing steps. A variety of discretization methods is available, some specifically tailored to aid in reverse engineering of logical models [55, 25]. In any case, the specifics of each method should be carefully noted and kept in mind when analyzing the resulting models.

Once the information is translated into a discrete or qualitative setting, different approaches allow further processing and deriving of parameter constraints. Formal methods from computer science, e.g., have garnered more and more attention regarding these tasks [30, 28]. Model checking techniques, which have proved greatly efficient for soft- and hardware verification, can be used to verify properties expressed as temporal logic formulas in complex state transition systems. Recently, they have also been exploited for the analysis of state transition graphs of biological systems [10, 29, 15, 9]. Computation tree logic, e.g., can be applied to non-deterministic transition systems such as asynchronous state transition graphs. A variety of interesting biological queries can be encoded, including the existence of specific state transitions, reachability properties or attractor characteristics. Highly efficient algorithms allow to check the parameter space for parameter sets specifying models with the desired properties. Generic formulas for queries of biological interest make the approach accessible for use [13, 41]. The method is not only suitable for inference of parameter constraints from given data, but also for checking consistency of the modeling assumptions and the available data, and for analysis of specified models.

The first chapter in this monograph, *Parameter inference for asynchronous logical networks using discrete time series*, introduces ideas for using model checking techniques in the manner described above, exploiting data given in the form of discrete time series. The method assumes that the modeler has already decided on the network structure and the range of the network components, i.e., the set of discrete values a network component may adopt, and uses

the Thomas formalism with the asynchronous update assumption. As described above, many specified models may be in agreement with the structural information, resulting in a pool of viable models that is often very large. The information to be exploited for validating the modeling assumptions and for finding parameter constraints yielding a more restricted model pool is given as a discrete time series. Such time series may have been obtained from quantitative time series via discretization, but they could also represent a series of qualitative observations over time.

For the purposes of the paper, a discrete time series is a matrix where the rows represent the subsequent measurements and each column is associated with a network component. The entries of a measurement may take values from the range of the corresponding component. For realistic use, they are also allowed to take the value -1, which is then neglected in the further analysis and may represent missing or questionable data.

The paper proposes several procedures for exploiting such a time series. At the core is a temporal logic formula testing the compatibility of a specified model with the time series. In essence, the formula encodes the existence of a path in the state transition graph traversing states that correspond to the measurements of the time series in the correct order. Note that if a measurement contains a -1 entry, then there is no unique state but a set of states that corresponds to the measurement. In a first step, the parameter space can be pruned by eliminating all parameter sets that do not generate such a path, i.e., that do not satisfy the formula.

The formula can be made stricter by specifying further properties for a path corresponding to the time series. The paper introduces the notion of monotonicity matrix, a Boolean matrix, whose columns again correspond to the network components and whose rows correspond to the transitions between measurements, i.e., it contains one row less than the time series. For each component, the value transition as indicated by subsequent measurements can be labeled as monotone by putting the corresponding entry in the monotonicity matrix to 1, if both measurement entries are not -1. This means, the value change of the component from one measurement to the next is supposed to either be a gradual increase or a gradual decrease, or it remains fixed when the values of both measurements coincide. In other words, there are no oscillations in the component value between the two measurements. If no such constraint is posed on a value transition, the entry in the monotonicity matrix is put to zero. The paper provides an algorithm to translate a time series and a monotonicity matrix into a corresponding temporal logic formula, the so-called monotone path formula.

Again this formula can be used to constrain the model pool, but it can also be a useful test for sufficiency of measurements in the time series. This becomes more clear, when keeping in mind that we can utilize model checking methods not only for deciding on one specified model but also for analysis of the incomplete information and modeling assumptions. The case that the model checking does not return a viable model, i.e., the model pool satisfying our specifications is empty, indicates that there are some inconsistencies in the modeling hypotheses. To exploit this fact, one could, e.g., take the strongest possible monotone path formula, which declares all valid measurement transitions to be monotone. If the remaining model pool is empty, there have to be some inconsistencies. Depending on whether the researchers place more confidence on the already specified model characteristics or on the sufficiency and correctness of the time series data, analysis can now focus on identifying where minor changes in the hypotheses result in a non-empty model pool. In the paper, we suggest some heuristics for pinpointing the problem in such cases, for changes in the network structure as well as weaknesses in the time series data. The results of such considerations can then be utilized for experimental design.

In particular, for revising the network structure, but also to extract as much information from the available data as possible, we introduce edge labels allowing to describe, or constrain, the effect an interaction has on the system's dynamics. The edge labels include constraints that test whether an interaction has an influence on the system's dynamics at all and also different levels of specifity for the character of such an effect, e.g., activating or inhibiting. The hierarchy of these edge constraints allows to finely tune the information inherent in the interaction graph and in turn to restrict the parameters consistent with this information.

Instead of ending up with no model fitting the constraints, or exactly one model which can then be used for further analysis, one often faces the problem that the model pool still consists of a great number of models. Information on the system can then be gained by characterizing the remaining model pool, e.g., by looking for parameter values coinciding in all models. Developing methods to extract meaningful information from the model pool certainly deserves significant attention. In the paper, we only touch upon ideas for this problem.

The techniques described above are illustrated using a small biological network, for which we took time series data from the literature. Since the data was quantitative, we used a standard discretization method to obtain a Boolean time series. Interestingly, analysis using the monotone path formula seemed to be able to indicate oscillations missed due to the discretization process and also a situation, where a multi-valued discretization would have been able to capture behavior the Boolean discretization eliminated.

The paper shows the potential of model checking techniques for the modeling (and the analysis) process, but also highlights some of the general difficulties of reverse engineering approaches. One of the main issues is certainly the fact that in most cases the available data can be used to constrain the model pool, but is by far not sufficient to specify a unique model. Consequently, the problem of finding meaningful and suitable criteria for selection of a model from the pool is of major interest. Although workable solutions exist, see e.g. [24], it is by no means solved. Furthermore, methods to extract characteristics of model pools will also contribute to the understanding of partially known systems.

In any case, methods for parameter inference need to be tuned to exploit the available data to the greatest possible extent. Apart from the goal of specifying a suitable model, this includes evaluation of the available data and extraction of problematic modeling assumptions accessible to further studies through experiments. Only experimental design tailored to the modeling purpose will result in efficient acquisition of useful data.

## Analysis

A specified model, consisting of an interaction graph and a corresponding fully-determined function, carries information on network structure as well as on dynamics. The network structure encoded in the interaction graph can be analyzed using graph theoretical methods [3]. As touched upon above, the interaction graph can carry varying amounts of information. In the simplest case, the graph is a directed graph just capturing the dependencies between network components. Often, a specific interaction can be characterized as activating or inhibiting, depending on the influence it exercises on the target component. This can be captured by labeling the edges in the interaction graph by a sign. However, in some cases the influence of an interaction may depend on the context, e.g., it may be activating in the presence of certain co-factors, but inhibiting in their absence. To indicate such properties in the network structure, it is represented as a multigraph

which allows for parallel edges with different labels. An even more refined representation is obtained by labeling edges not only with a sign but with information on when the edge becomes active. This depends on the activity level of the tail vertex of the edge. In the multi-valued Thomas formalism, e.g., edges are labeled with a threshold value. If the activity level of the tail vertex crosses the threshold, the edge becomes active. In summary, interaction graphs come in many different flavors. Structural analysis of the network is then based on the type of interaction graph, and may incorporate in addition to purely topological features the information inherent in the edge labels.

In a well-defined model, network structure and dynamical behavior should be consistent. In other words, all interactions in the structure and their labels should be recoverable from the network dynamics. This is of particular importance if relations between structural and dynamical characteristics are to be investigated. Checking structural and dynamical consistency can also be seen as a first step in model analysis, since it verifies the compatibility of the underlying modeling assumptions. Model checking techniques can be used for this task, as described above. A different possibility is to derive an interaction graph directly from the update function. Dependencies between components and interaction character can be extracted from the discrete derivative [50, 48]. The resulting graph then represents the structure underlying the update function and should match the interaction graph.

In general, the structural information is not sufficient to understand the behavior of the system encoded in the state transition graph. Analysis of this graph is often impeded by its size, which is exponential in the number of network components, and, at least in the asynchronous case, its complexity. As for quantitative models, simulation can be a useful tool to understand the behavior of the system given initial conditions. In the case of asynchronous dynamics, specification of an initial state does not necessarily result in a unique trajectory but may restrict the dynamics to a smaller subgraph. A more comprehensive analysis is possible by using model checking techniques. As described above, they allow for efficient analysis using suitable queries encoding dynamical characteristics.

A different possibility is to derive a more compact representation of the state transition graph that retains information of interest but is more accessible for analysis. Modularity approaches reduce complexity by representing subgraphs with certain properties as one vertex in a new quotient graph [39, 63]. An often used representation following this idea is the so-called strongly connected component graph, where every strongly connected component is represented by a single vertex while edges from vertices from one component to vertices in another component are merged. The result is an acyclic graph, where the attractors are easily identified as the vertices with out-degree zero.

Modularization methods can also be applied to the system itself rather than the state transition graph. They try to break the network down into building blocks aiming for model reduction, but also for a deeper understanding of the way a system's structure and function can be characterized by its substructures. The idea of modularization is well-established in systems biology, but approaches are by no means homogeneous. They range from biologically motivated ideas of system modules to purely mathematical approaches based mainly on the interaction graph [33, 44, 45, 4]. Clearly, modularization solely based on network structure is not ideal when the research interest lies in understanding a system's function. In order to discover subnetworks of significance for the system's behavior or exploitable for model reduction, information about the system's dynamical characteristics has to be invested [12, 26, 51].

In a series of papers, which are presented in the second section of this monograph, I have developed modularization methods for discrete dynamical models yielding network modules on the basis of structural and restricted dynamical information. Such modules consist of an interaction graph, which is a subgraph of the interaction graph of the original system, and an update function which, roughly speaking, represents the dynamics of the original network projected on the components of the module's interaction graph. Such modules can be utilized for model reduction purposes. In addition, they clarify the relation between structural and dynamical properties. For example, I characterize modules responsible for number and shape of attractors using the module interaction graph.

The core observation motivating the definition of such modules is a very simple one. If a given network decomposes into independent components, i.e., the interaction graph is not connected, then those components can be easily analyzed in isolation from each other. That is, we can consider smaller networks for which the interaction graph corresponds to one component of the original interaction graph. Since there are no dependencies between the different interaction graph components, each coordinate function belonging to a vertex in such a component does not depend on the values of vertices outside the component. We can then define the smaller system, i.e., the network module, via the corresponding coordinate functions of the original system, and consider the dynamics on a smaller state space, namely the projection of the original state space on the module components. To obtain the state transition graph of the original system from the modules, we just have to consider the product space of the module state spaces, and preserve the edges of the module state transition graphs in a manner that depends on the considered type of update, synchronous or asynchronous. In any case, we can obtain the complete state transition graph of the original network from the modules in a straightforward procedure.

The above observation is of very limited use, since the networks we are dealing with in systems biology are generally connected. However, the properties of the modules described above are highly desirable. They allow, on the one hand, for efficient analysis of the system, since the behavior they show in isolation is preserved, and easy procedures allow to reconstruct the dynamics of the original network from the module dynamics. On the other hand, a much clearer correspondence between dynamical and structural characteristics of the network can be obtained, since a module interaction graph represents the structural aspects underlying the dynamical features of the original network generated by the module. The question now is whether it is possible to find such network modules in a, maybe even densely, connected network.

The central idea to approach the problem is to identify modules that behave as if isolated, if not in the entire state space, then at least locally. To achieve such a local isolation, we focus on components within the network whose value remains fixed along all trajectories traversing some region of state space. Their importance for network analysis has already been recognized by Kauffman and colleagues, who called them frozen components [35]. One most useful property of such components that remain essentially dynamically inert is the following. Since we already observed that the component does not change its value on trajectories in the designated region of state space, we know that the behavior of other components within that region does not influence the frozen component, i.e., locally there is no dependency between them. On the other hand, the influence exhibited by the frozen components is a constant one. This in turn means, that we can basically eliminate all interactions involving the frozen component, and eventually drop the frozen component itself, from a representation of the network structure of importance for the behavior in the region. The resulting local interaction graph is a subgraph of the original interaction graph, but may include significantly fewer edges and thus may be disconnected.

The connected components can then again be seen as interaction graphs of network modules. The respective update functions are constructed from the corresponding coordinate functions, where, other than in the isolated case, the values of the frozen components have to be taken into account. Those values can be interpreted as boundary conditions for the network module. The resulting modules show the desired properties for network reconstruction, however, they only yield information on the original network restricted to the considered region of state space.

The core lemma of all the papers included in the second part of the monograph formalizes the above observation in different settings, establishing a way to recover isolation locally using frozen components. The property of a frozen component regarding its behavior is rephrased as fixed point property of a suitable set function. More precisely, a function is defined on a set of so-called singular or symbolic states, each of which represents a set of states. Given a fixed point of this function, called singular or symbolic steady state, each of its singleton components represents a frozen component which is dynamically inert in the region of state space represented by the steady state. Due to the local isolation, the local dynamics of the original system can be reconstructed from the resulting network modules. The series of papers elucidates a number of mathematical properties related to symbolic steady states, using methods from Boolean spectral and iteration theory, and develops a modularization technique based on the results. As mentioned, all papers overlap in their observations on locally isolated modules. In the following, the development and the progress from paper to paper, as well as the different problems addressed, are detailed.

The focus of the first paper, *Relating attractors and singular steady states in the logical analysis of bioregulatory networks*, was not yet on network modularization, but aimed at transferring the notion of singular steady state introduced by Snoussi and Thomas for multi-valued logical models to Boolean networks [59]. Underlying this notion is the idea that each interaction in the network corresponds to some threshold, i.e., the interaction will become effective once the activity of its tail vertex is above a certain threshold. In the logic-based formalism, the threshold itself is not explicitly modeled. There is only an activity level above and one below it. Snoussi and Thomas propose to consider both activity levels framing a threshold, and compare the impact of both choices on the network dynamics. They thus extend the state space to include states containing threshold values, which are called singular as opposed to regular values. The dynamics in states with singular components, called singular states, is then determined by considering the adjacent regular states, i.e., states that for the singular components only contain activity levels just above or below the corresponding threshold.

Snoussi and Thomas mainly introduce singular states to determine the impact of feedback circuits, i.e., cycles in the interaction graph, on the dynamical behavior of the system. Such a circuit is called positive, if the number of negative edges in the cycle is even, and negative otherwise. Thomas conjectured more than thirty years ago that there has to be a positive circuit in the interaction graph, if the system has multiple attractors, and that a negative circuit is necessary for the system to generate a cyclic attractor, i.e., an attractor of cardinality greater than one. Both conjectures have been proven for discrete networks using the asynchronous update assumption [60, 49, 47, 48]. Easy examples show that the existence of the respective circuits is not a sufficient condition for the described attractor characteristics. Networks consisting of a single circuit do display these characteristics, but the property may be lost when the circuit is embedded in a more complex network. Snoussi and Thomas use the notion of singular state to address this question and propose a method to test the functionality of a circuit, i.e., its poten-

tial to imprint its characteristic isolated behavior on the network dynamics, based on analysis of singular steady states whose singular components correspond to the circuit interactions. Their results rely heavily on the assumption that for each component all outgoing edges in the interaction graph are labeled with distinct thresholds, i.e., there is a unique correspondence between the singular values and the interactions. Moreover, they restrict the class of functions they consider for a given interaction graph, e.g., not allowing the character of an interaction, activating or inhibiting, to depend on the state of the system.

In our paper, we adapted the notion of singular states to be suitable for Boolean networks, a framework in which we clearly cannot maintain the condition of distinct thresholds for interactions originating in the same component. In fact, in a Boolean network all thresholds are by definition between the values zero and one. In agreement with Snoussi and Thomas, we consider asynchronous update and restrict the class of suitable Boolean functions to those that do not allow for ambivalent interaction character, the so-called sign-definit functions. We define a function that allows us to include singular states in the dynamics, again based on comparison of the function values of corresponding regular states. The focus of the paper is on singular steady states and their relation to attractor properties, aiming at an adaptation of the results of Thomas and Snoussi.

As a first result, we provide a characterization of singular steady states using certain adjacent regular states. This allows for the identification of singular steady states solely using the original update function. Then we focus on properties of singular steady states. The values of the regular components, i.e., those not associated with a threshold value, of a singular steady state remain unchanged, independent of a choice of regular value for the singular components of the state. Thus, they signify frozen components for the subspace of state space determined by all regular components of the singular steady state. We define a local interaction graph associated with a singular steady state that contains only the dynamically active components to derive subnetworks or modules that govern the behavior of the original system in the considered subspace, as described above. Based on the resulting network decomposition, we prove the core lemma concerning the quasi-isolation of the network components corresponding to the singular values. We then show that the attractors of the original network can be composed from those of the network modules consisting of the local interaction graph components and suitably defined update functions.

Having this tool in hand, we focus again on feedback circuits. We prove that a module whose interaction graph contains only one circuit, positive or negative, generates the corresponding attractor characteristics in the original network. The proof is constructive for the attractors of the original system. We clarify the impact of embedded circuits further by providing counterexamples that show that the statement is generally not true if we allow for the module to contain more than one circuit, and that the sufficient condition for multiple resp. cyclic attractors we proposed is not a necessary condition.

The framework and the results are significantly extended in the second paper, *Deriving behavior of Boolean bioregulatory networks from subnetwork dynamics*. The focus is still on singular steady states of Boolean networks. However, all constraints restricting the class of Boolean functions to be considered are dropped, i.e., the theory is extended to Boolean functions in general. I still use a framework based on the Thomas formalism, i.e., a network is described by its interaction graph and a set of logical parameter values depending on the network structure specifying the update function. However, we now allow for parallel edges of different signs in

the interaction graph, reflecting context sensitivity of interactions. The dynamics is represented as asynchronous state transition graph. Not restricting the class of viable Boolean functions amounts to dropping certain constraints on the logical parameters, but it still has to be ensured that the dynamical behavior reflects the underlying network structure, i.e., existence and character of interactions have to be inferable from the state transitions. This is translated again into constraints for the logical parameters. However, these constraints do not exclude any Boolean function, but only ensure the consistency of update function and network structure. The proof of this statement is not given in the paper, but can be found in a related conference paper of mine [56].

The more general framework necessitates an adaption of the update function used for singular states. Once this is provided, the focus is again on singular steady states. The method to derive network modules crucial for the dynamics in a corresponding region of state space is refined. The definition of the so-called product state transition graph captures the procedure necessary to derive the dynamics of the original system from the network modules. It is shown that the product state transition graph corresponds to the state transition graph of the original system restricted to the states in the affine subspace represented by the singular steady state. An easy corollary allows again to match the attractors of the original system to the attractors of the modules, and to generalize the results concerning sufficient conditions for certain attractor characteristics based on the existence of a unique circuit in the interaction graph of a module.

In the last part of the paper, I address two questions of interest for the application of the results. The first concerns the suitability of the local analysis of a system via modules derived from singular steady states for a comprehensive analysis of a given system. Conditions, e.g., that ensure that the module analysis of a specific set of singular steady states renders all attractors of the original systems, are of great interest. Secondly, it is self-evident that an efficient method for identification of suitable singular steady states has to be provided in order to make application feasible.

Both issues can be successfully solved for a certain class of networks called networks with input layer. Such networks contain components whose update function is the identity. Thus, their values remain fixed on all trajectories, i.e., they are frozen components. Such components are often used to model stable environmental conditions for a network, or incoming signals, and are called input vertices. Fixing the values of the input vertices amounts to imposing constraints on the systems dynamics. Iteration of the singular state consisting of the chosen values for the input vertices and all singular values otherwise results in constraint propagation through the network. The iteration process terminates in a singular steady state. It can be easily seen from the proof that this process takes at most $k$ steps, where $k$ is the number of network components that are not an input vertex. Using this procedure, each combination of values for the input vertices can be associated with the resulting singular steady state. Concluding the paper, I show that all attractors of a network with input layer can be derived from the modular analysis using this set of singular steady states.

Two further articles based on the ideas and questions raised in the previously described paper are included in this monograph. The first, *Analysis of discrete bioregulatory networks using symbolic steady states*, is a journal publication with a rather long research and review process, while the second, *Dynamical and structural modularity of discrete regulatory networks*, was a conference contribution. Although the journal article is the more recent publication, I placed it before the conference paper, since work on it started earlier and some of its results were the

basis for the conference paper.

In the journal paper, I make the step from a purely Boolean formalism to a multi-valued one. To simplify the mathematical definitions and proofs, I chose to represent the network by the update function and derive local and global interaction graphs using its discrete derivative rather than defining first the interaction graph and parameter sets and adding several constraints guaranteeing the consistency of structure and dynamics. The resulting interaction graphs are still signed multigraphs. In particular, they carry no information on thresholds. In that sense, the interaction graphs in the multi-valued case carry less information than in the comparable Boolean setting. In the paper, I leave the interpretation of singular values as threshold values as motivated by Snoussi and Thomas behind. Instead, I focus on the singular states being a symbolic representation of a set of regular states. In the multi-valued formalism, I extend the set of regular states by considering so-called symbolic states encoding the regular states of an affine subspace of the state space. An adapted update function for the symbolic states is introduced, which allows the definition of a symbolic steady state as a fixed point of this function. Apart from making the non-trivial step from Boolean to multi-valued functions, I also include for the first time a thorough investigation of the synchronous besides the asynchronous dynamics.

The results from the Boolean setting concerning the network modularization and reconstruction of the original dynamics can be transferred in their entirety to the multi-valued setting. Similar results are obtained for synchronous dynamics, however, the reconstruction process yields slightly different specifics for the product state transition graph. The differences are highlighted by statements giving the number and size of attractors depending on the number and size of the corresponding module attractors.

Based on these results, I aimed for a generalization of the statements linking positive and negative circuits to attractor number and size for asynchronous and synchronous dynamics. The multi-valued situation concerning this issue is more delicate than in the Boolean case. There is no information on the values that allow an edge in a circuit to be active and neither on whether the activation of such an edge will lead to a value change of its successor that in turn is crucial for activity of the successor edge in the circuit. Using this fact, we can easily construct examples of multi-valued networks containing circuits in the interaction graph that show none of their characteristic behavior. The properties concerning the activity of circuit edges just described have to be explicitly required in order to transfer the results concerning circuits in network modules and their impact on the attractors of the original system from the Boolean to the multi-valued setting. This links back to the important question of feedback circuit functionality shortly mentioned above. Provided the requirements are met, the results are valid in the asynchronous as well as in the synchronous case.

Following the results illustrating the usefulness of symbolic steady states for network analysis, the paper thoroughly discusses identification methods for symbolic steady states. Motivated by the ideas for networks with input layers, the constraint propagation procedure via iteration is applied to states containing frozen components. Each such state, called seed, can then be associated with the symbolic steady state resulting from constraint propagation. Modular analysis using the symbolic steady state does not only cover the asymptotic behavior in the subspace corresponding to the symbolic steady state, but in the potentially much larger one corresponding to the seed.

States with regular values for input vertices then represent a special case of seeds, but the underlying characteristics allowing the identification of seeds can be generalized significantly. For this purpose, I discuss in the paper the notion of canalyzing function. A component update

function is called canalyzing, if there exists a predecessor of the component in the interaction graph and a value, called canalyzing value, for that predecessor component such that the image of the component update function, called the canalyzed value, is already determined by this specific predecessor value. In other words, this particular predecessor is the defining influence once it adopts the canalyzing value. Since the canalyzed value could in turn be a canalyzing value for some other component in the network, a canalyzing function may impact several network components. In this sense, canalyzing functions can interlock along paths in the network structure, which results in a stabilizing effect on the network dynamics. If such paths constitute a circuit, this effect generates seeds, namely the states where the circuit components adopt the corresponding canalyzing values. In consequence, identification of seeds can be based on structural analysis determining the circuits in the interaction graph and analysis of the corresponding update functions.

In the last section of the paper, the methods and results concerning symbolic steady states are applied to a logical model of T helper cell differentiation taken from the literature. It consists of 17 components, three of which are input vertices. The proposed methods result, on the one hand, in a significant reduction of analysis complexity. On the other hand, a clearer understanding of the relation between the structural design of the network and its functionality is achieved through the refined view offered by the network modules.

All papers mentioned so far in this section have been set in the context of network modularization, however, as already noted, this was not the original motivation. Once the potential of the approach as modularization technique became evident, I tried to put it more rigorously into context. As already stated, there is no clear and unique definition of a network module, since different authors focus on different aspects characterizing a module.

The last paper included in this section, *Dynamical and structural modularity of discrete regulatory networks*, has two main parts. On the one hand, it establishes a refinement of the modularization technique presented in the previous paper for multi-valued networks with asynchronous dynamics. On the other hand, it provides concise terminology and a mathematical characterization for the modules resulting from the symbolic steady state analysis.

The idea of the refined approach originated in the analysis of the previously mentioned T helper cell differentiation model, where the extended method renders stronger results. Basically, rather then restricting the symbolic representation to affine subspaces, the new method also allows cartesian products of arbitrary discrete intervals in the component ranges. The specifities of the asynchronous update require careful consideration when adapting the update function for the new set of symbolic states.

The finer grained symbolic states allow for an easy procedure to identify for a given regular or symbolic state a hyperrectangular region of state space which contains all trajectories starting in that state, or starting in the set of states corresponding to the symbolic state, respectively. It is based on an iteration process, but again has to take into account the particularities of asynchronous update. The resulting region can be represented by a symbolic state which is called extended forward orbit of the state under consideration. The regular components of such a symbolic state can be interpreted as frozen components of the initial regular or symbolic state. Further iteration then leads to a symbolic steady state. To set the result into context with the previously discussed paper, the extended forward orbit can be seen as a seed in the more general setting. Further core results concerning the modularized analysis carry over to the more general setting. The paper illustrates the approach using again the T cell differentiation network.

As mentioned, apart from the extension of the framework, the article aims at rendering the notion of module more precise. Motivated by the broad range of available modularization techniques focussing on a variety of aspects, I differentiate between structural, dynamical and network modules. Both the structural and the dynamical modules are defined only in the broadest sense, a structural module being a subgraph of the interaction graph, and a dynamical module being either a subgraph of the state transition graph or the projection of such a subgraph to a smaller set of components. Both definitions serve to illustrate the link of purely structural and purely dynamical aspects to the network modules. These are the modules of interest for modularized analysis that builds upon suitable subsystems to obtain information on the dynamics and the links between structural and dynamical characteristics of the original system.

I define a network module as an update functions on a set of states, which is derived from a subset of the original state space by projection on a set of components, such that the module update function captures the dynamics of the original function on that component subset. A network module is called autonomous if the range of its update function is contained in its domain, so that iteration and thus deriving dynamical behavior becomes possible. An autonomous network module can be treated as any regulatory network. The interaction graph of the network modules is then a structural module of the original system, and its state transition graph is a dynamical module of the original system. With this terminology in hand, the analysis based on symbolic steady states can be clearly categorized as modularization technique yielding network modules.

The results of the papers described above offer methods for efficient analysis, but also help to clarify the relation between structural and dynamical characteristics of discrete networks. A particular focus is on functional circuits in the interaction graph. Their importance for characteristic dynamical behaviors is well known, but not yet thoroughly exploited. Other approaches focussing on conditions that allow circuits in the interaction graph to imprint their behavior on the network dynamics dovetail nicely with the ideas presented here [43, 46]. The results indicate the possibility to exploit positive circuits to determine a discrete analogon of a separatrix, while negative circuits shape the complexity of the behavior in the different regions of the resulting state space partition. The ultimate goal is to obtain a comprehensive understanding of the global dynamics from a circuit analysis of the interaction graph, i.e., to define an analysis method that does not require to explicitly generate the state transition graph.

As seen, the results of the papers are not restricted to circuits in the interaction graph. Similar ideas can be used to identify more general functional modules that transfer their characteristic behaviors to the entire system. Of particular interest are modules of known biological importance, such as coupled feedback circuits and feedforward loops. After characterizing their behavior in isolation, the aim is again to identify corresponding functional modules in complex networks, leading to a network modularization integrating structural and dynamical aspects. Such a modularization could, on the one hand, highlight the relations between structural network design and system's function, which is of central interest for biological networks. It could also provide further strategies to restrict the model pool resulting from reverse engineering problems, and highlight structures of particular interest for experimental design. On the other hand, it will be exploitable for model reduction purposes, again resulting in efficient analysis methods.

Lastly, it needs to be mentioned that the results of dynamical analysis strongly depend

on the chosen modeling formalism. When talking about module functionality, e.g., respective qualitative and quantitative representations may lead to essentially different features. Thus, evaluation of the results of network analysis should always take the particularities of the underlying formalism into account.

### Refinement

Network analysis aims at reaching a clearer understanding of the modeled system and ideally results in focussed suggestions for further experiments. Newly available data expands the basis for modeling and usually necessitates tuning of the original model. The process is then repeated, each iteration leading to a more realistic model. However, the modeling formalism chosen in the beginning may at some point not be suitable anymore to capture all the information inherent in the available data. The need to evolve a model across the boundaries of modeling formalisms, e.g., from a purely discrete to a quantitative or stochastic model, generates a number of interesting theoretical and practical challenges.

A first question concerns methods to transfer a model from one formalism to another. Several, partly generic, approaches exist to generate differential equation models from discrete models and vice versa [20, 65, 18]. Hybrid methods have been developed to bridge the gap between two modeling paradigms. There are several frameworks available that still have an underlying discrete model description, but enrich it with quantitative or stochastic aspects, e.g., Petri net, process algebra or hybrid automata formalisms [6, 30, 14].

Coupling of different model types to generate a comprehensive model has been considered in the context of constraint-based modeling of metabolic networks coupled with regulatory and signal transduction processes [16, 17, 40]. Mainly, they only allow for a very restricted dynamic evolution of the coupled system, and the problems arising when coupling models using different representations of time, e. g., continuous and discrete, are by no means solved. Hybrid formalisms may help finding solutions to integrating static, discrete and continuous models.

To evaluate the applicability of such methods on a theoretical level, a comprehensive mathematical understanding of the relation between different formalisms is indispensable. Piecewise affine differential equations, e.g., offer a compromise between discrete and ordinary differential equation models. They help to understand consistencies and differences between the dynamics generated by discrete functions and differential equation models, respectively [32, 20]. Further research in this direction is needed to detect the impact of different modeling strategies on analysis results. This will not only drive the development of formalisms offering a seamless integration of new data, but will also allow for a fruitful complementary use of different formalisms. Similar network characteristics observed for corresponding models in different formalisms may potentially be intrinsic features of the underlying biological system.

This last issue is the focus of the first chapter in this section, *Comparing discrete and piecewise affine differential equation models of gene regulatory networks*.

It has long been known that asynchronous logical models, and, in particular, the Thomas formalism allowing for multi-valued logical modeling with asynchronous update, is in good correspondence with differential equation systems consisting of sigmoidal, or Hill, functions [31, 58]. The steeper the slope of such a function, reflected in the size of the Hill coefficient, the better is its correspondence to a step function, which can be seen as the basis for a Thomas model. Piecewise affine differential equation (PADE) models can be seen as an intermediate between

the differential equation model using Hill functions and the purely discrete Thomas model, since they use step functions over continuous time. In a series of papers, de Jong and colleagues proposed an approach for modeling gene regulatory networks with PADE models [21, 20, 8]. They presented efficient analysis methods for such models based on a corresponding qualitative representation. This representation is not dependent on the precise parameter values of the PADE but rather on certain constraints on those values, and works on a partition of state space whose domains are represented as discrete states. The dynamical behavior is captured in a directed graph, called qualitative transition graph, which is shown to be an overapproximation of the behavior of a corresponding PADE system.

The qualitative representation of a PADE system by de Jong and colleagues is clearly closely related to the Thomas formalism. A PADE system can easily be translated into a discrete update function and vice versa. However, the qualitative transition graph takes into account the threshold values of the step functions in the PADE. The behavior for states including a threshold value is derived from the properties of the adjacent regular states, i.e., states not containing a threshold value. This particularity complicates the comparison between the asynchronous state transition graph and the qualitative transition graph of corresponding discrete and PADE models. Available results, as mentioned in the paper, mostly focus on specific dynamical properties, and are partly only stated for Boolean systems. A clearer understanding of the correspondences would allow for a more targeted choice of modeling framework depending on the system to be modeled and the question to be investigated.

In the article, we aim at clarifying the relation between the two different formalisms based solely on the information inherent in the state transition and the qualitative transition graph, respectively. First, we establish the correspondence between a set of PADE systems characterized by certain parameter constraints and a suitable update function of a Thomas model. In the following, we focus on the two graphs representing the dynamics. The core result characterizes edges in the qualitative transition graph using sets of edges in the state transition graph. While the statements are straightforward for domains not including threshold values, they become much more involved if threshold values are included. In consequence, the proof, although basically a translation of the requirements ensuring a transition in the qualitative transition graph into the discrete setting, becomes quite technical.

The use of the theorem relating the edges is two-fold. On the one hand, it can be utilized as a tool for proofs relating more complex features of the two graphs, such as paths or attractors. On the other hand, it provides a starting point for the construction of counterexamples in the cases where the correspondences between the dynamics break down. Both applications of the theorem are illustrated in the article. It becomes apparent that, although features involving only regular domains and domains associated with a single threshold value are mostly preserved, many important dynamical characteristics, such as reachability properties or cyclic attractors, do not necessarily coincide. In particular, this shows that the qualitative transition graph does not simply constitute a refinement of the state transition graph produced by the inclusion of threshold values. Understanding the characteristics of networks with corresponding qualitative and state transition graphs could help to clarify the relation between the two formalisms.

The next paper included in this section, *Temporal constraints in the logical analysis of regulatory networks*, offers a different approach to integrating a continuous time evolution in an inherently discrete dynamical model. The framework, which we first introduced in 2006, aims at enriching a Thomas model with information pertaining the duration of the component value update

processes involved. As described above, the asynchronous update assumes that the update of a component value is associated with a specific time delay depending on the biological process represented by the component update, and that all the time delays differ. Since no further information about the time delays is included in the model, this assumption results in the non-deterministic asynchronous state transition graph. If we could integrate potentially available information on the time delays, we might be able to eliminate certain state transitions and obtain a more concise representation of the dynamics.

We chose to base our hybrid framework combining a Thomas model with continuous time evolution on the theory of timed automata [5]. Within this framework, discrete events can be associated with clocks measuring time. There exist several modeling and analysis tools for timed automata models which we can exploit for our approach. In particular, timed automata models are accessible with model checking methods.

A network in the formalism is specified component wise, which allows for easy model refinement and extension. Each network component is modeled separately as a timed automaton, which carries information on the value range and the update function of the corresponding component. Also included are representations of the update processes, increasing and decreasing, which are called intermediate locations. Once the component enters an intermediate location, a clock starts measuring time. If the corresponding time delay has passed, the component is forced to complete the value change. We do not require the time delay to be specified exactly, but rather allow to indicate an upper and lower bound for the time delay, acknowledging the difficulty of obtaining such information experimentally. This again results in a possibly non-deterministic representation of the dynamics. Once the components of the network are modeled, the transition systems representing the dynamics of the entire network, taking into account the discrete update rules as well as the constraints imposed by the time delays, can be easily and automatically generated.

Before illustrating the potential of the new approach on a small example, we clarify the relation between the hybrid and the Thomas formalism. We show that, provided suitable time delay values are chosen, the hybrid model can reproduce the dynamics of the underlying Thomas model. Since the transition system of the timed automaton model is described on states that may include intermediate locations, the correspondence between the transition system and the state transition graph of the Thomas model is not immediately evident. However, we develop a meaningful way to relate paths in the transition system to paths in the state transition graph. Based on the this idea, we can extract a quotient graph of the transition system which is isomorphic to the corresponding state transition graph. Consequently, the hybrid modeling formalism can be seen as a refinement of the Thomas formalism.

The last section of the article illustrates the modeling process and the advantages of the refined representation on a model of the bacteriophage $\lambda$. The resulting model is implemented in a software suitable for timed automata models, which allows for analysis using model checking. Translating information from the literature on the sequence of gene expression after a specific signal into time delay constraints already yields a much clearer picture of the dynamical behavior than that encoded in the purely discrete state transition graph, since several state transitions are not in agreement with the time constraints. In addition, observed behavior, e.g., oscillations represented by cycles in the state transition graph, can now be analyzed with respect to stability. In some cases, analysis including the time delays shows that trajectories cannot remain in a cycle indefinitely, or that the constraints on the choice of time delay values that allow the system to remain in a cycle are very strict and thus susceptible to perturbation. In other cases, trajectories

may travel along a specific path regardless of the time delays chosen, i.e., the behavior of the system in a corresponding state is not influenced at all by fluctuation in the durations of the processes involved.

The results of the paper show the potential of hybrid methods, which are more expressive than a purely discrete approach. The framework introduced above can easily be extended to capture further details about the modeled system, as shown, e.g., in a continuative publication not included in this monograph [57]. In addition, other hybrid discrete/continuous approaches utilizing the theory of hybrid automata have been introduced in recent years [2, 7]. However, the increased expressiveness has to be paid for by additional parameters for the continuous processes and a more involved analysis of the resulting model. Since parameter inference is already a significant difficulty in the purely discrete setting, research should focus not least on efficient analysis methods applicable to partially specified models.

The last paper included in this section, *Stochasticity in reactions: a probabilistic Boolean modeling approach*, introduces a hybrid formalism combining logic-based modeling with stochastic aspects. It was motivated by and developed during interdisciplinary research on a signaling network in plants.

The underlying discrete formalism allows for a local modeling process, i.e., rather than defining update functions for all components from which we then can derive the state transitions, we describe the different processes making up the behavior of the system separately. That is, we first specify the set of network components and then define a set of so-called reactions. A reaction consists of a vector describing the effect execution of this reaction has on a given state. For example, to model degradation of a substance we would describe its effect by a vector consisting of a -1 entry for the component representing the substance and zero entries otherwise. In addition to the effect vector, a reaction is characterized by a function giving the probability of the reaction occurring depending on the state the system is in. This function also guarantees that reactions occur in suitable states, e.g., degradation of a substance will only take place if that substance is present. In the paper, we only allow for the function to assign either probability zero or some predefined constant. However, extension of the framework to include a more sensitive dependency of the reaction probability on the state would be straightforward.

Reactions may effect more than one component. If a reaction is executed, all components effected by the reaction are updated in concert. In addition, several reactions may, but do not have to, take place simultaneously. The probability of such an event is being calculated as a joint probability. All possible state transitions and their respective probabilities are summarized in a probabilistic state transition graph. This graph in general neither corresponds to a synchronous nor to an asynchronous state transition graph, since closely related updating processes may be combined in one reaction but different reactions or sets of reactions may by executed in a given state resulting again in a non-deterministic state transition graph. The dynamics of the network can be represented as a Markov chain, making it accessible to the corresponding analysis methods and tools.

The resulting framework is then put into context with regards to other approaches using stochastic Boolean networks. We exemplarily perform a detailed comparison of our framework with the probabilistic Boolean network (PBN) approach by Shmulevich and colleagues [54]. We show that each PBN model can be captured in our formalism, while the reverse statement is only true if reactions influencing the values of more than one component are excluded. A slight generalization of our approach allows for a much stronger statement. Given an arbitrary

probabilistic state transition graph on a Boolean state space, we can specify a model reproducing that graph.

As mentioned, analysis can make use of theory and tools concerning Markov chains. In addition, we introduce some ideas to analyze models with incomplete information on the stochastic parameters, i.e., on the specific values to be assigned as probability values of the reactions. We propose to use optimization to derive parameters best suited for a predefined purpose, such as ensuring that the system reaches a specific attractor. Analysis can also be restricted to subgraphs of the probabilistic state transition graph, which in turn might be identified using the underlying discrete network.

The formalism and analysis methods are illustrated on the cytokinin signaling network in *Arabidopsis thaliana*. Together with plant biologists, we try to elucidate the specifics of a certain negative feedback mechanism in the signaling process. The modeling formalism described above was directly motivated by this research, since earlier work indicated that purely discrete modeling was not sufficient to capture essential characteristics of the feedback mechanism and effect. Analysis of the hybrid discrete/stochastic model indicates a dependency of the feedback mechanism on the degradation rates of one of the substances involved, and thus offers an experimentally verifiable hypothesis.

The integration of stochastic effects in the discrete dynamics allows for flexible and often more realistic modeling. The reaction probabilities are not only suitable to capture natural fluctuations in biological processes but can also be used to model time delays, where a high probability may represent a short time delay. But again, the extension of the discrete formalism results in additional parameters that may be hard to determine. Consequently, analysis of only partially specified models is of interest here as well. Efficient methods should exploit the underlying discrete network in order to allow for a more targeted analysis of the probabilistic state transition graph or to identify smaller network modules the analysis may focus on.

As shown for the specific hybrid formalisms mentioned above, good expressiveness within a modeling framework is desirable to encode all known system specifics, but it should always be weighed against the price one has to pay in terms of additional, and often hard to determine, parameters and analysis costs. A clear understanding of the relations between different formalisms will, on the one hand, allow for a well-founded choice of modeling formalism for a given system. On the other hand, it may open up possibilities to exploit analysis of corresponding lower-level models to reduce the analysis complexity of a more refined model. In addition, a more abstract view on a system may offer insights about essential characteristics of a system that are much harder to discern in a more detailed model. In light of these considerations, a gradual model refinement process, which takes careful note of the model development across the boundaries of modeling formalisms and keeps exploiting the available coarser models, may be particularly fruitful.

All three research areas presented above generate a variety of interesting research questions, on the mathematical as well as on the application side. In fact, as seen, biological questions generate mathematical challenges. Results can then be applied and tested for their significance in a real-world setting, leading to new ideas concerning targeted mathematical methods. In this respect, interdisciplinary research in systems biology creates synergies, allowing it to be more than the sum of its parts.

*Organizational remarks:* The contributions to each of the three fields mentioned above are presented as chapters of this monograph. The chapters start with some general information on the authors of the respective material as well as a statement of my contribution to the results. In most cases, the chapters correspond to papers published in journals or conference proceedings, the details are also given in the preliminary remark. All sections of the original articles are presented, including introduction, preliminaries, conclusion and bibliography. In particular, relevant terminology and notation are introduced in each chapter.

## References

[1] W. Abou-Jaoudé, D. A. Ouattara, and M. Kaufman. From structure to dynamics: Frequency tuning in the p53-mdm2 network: I. logical approach. *J. of Theor. Biol.*, 258(4):561 – 577, 2009.

[2] J. Ahmad, G. Bernot, J.-P. Comet, D. Lime, and O. Roux. Hybrid modelling and dynamical analysis of gene regulatory networks with delays. *Complexus*, 3(4):231–251, 2006.

[3] R. Albert. Network inference, analysis, and modeling in systems biology. *Plant Cell*, 19(11):3327–3338, 2007.

[4] U. Alon. Network motifs: theory and experimental approaches. *Nat. Rev. Genet.*, 8:450–461, 2007.

[5] R. Alur. Timed Automata. In *Proceedings of the 11th International Conference on Computer Aided Verification*, volume 1633 of *LNCS*, pages 8–22, 1999.

[6] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas. Discrete abstractions of hybrid systems. In *Proceedings of the IEEE*, 2000.

[7] G. Batt, R. Ben Salah, and O. Maler. On timed models of gene networks. In J.-F. Raskin and P. Thiagarajan, editors, *Formal Modeling and Analysis of Timed Systems*, volume 4763 of *LNCS*, pages 38–52. 2007.

[8] G. Batt, H. de Jong, M. Page, and J. Geiselmann. Symbolic reachability analysis of genetic regulatory networks using discrete abstractions. *Automatica*, 44(4):982–989, Apr. 2008.

[9] G. Batt, M. Page, I. Cantone, G. Goessler, P. Monteiro, and H. de Jong. Efficient parameter search for qualitative models of regulatory networks using symbolic model checking. *Bioinformatics*, 26(18):i603–i610, 2010.

[10] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. Application of formal methods to biological regulatory networks: extending Thomas' asynchronous logical approach with temporal logic. *J. Theoret. Biol.*, 229(3):339 – 347, 2004.

[11] S. Bornholdt. Less is more in modeling large genetic networks. *Science*, 310(5747):449–451, 2005.

[12] F. J. Bruggeman, H. V. Westerhoff, J. B. Hoek, and B. N. Kholodenko. Modular response analysis of cellular regulatory networks. *J. Theoret. Biol.*, 218(4):507 – 520, 2002.

[13] N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter. Modeling and querying biomolecular interaction networks. *Theoretical Computer Science*, 325(1):25 – 44, 2004. Computational Systems Biology.

[14] C. Chaouiya. Petri net modelling of biological networks. *Brief. Bioinform.*, 8(4):210–219, 2007.

[15] F. Corblin, E. Fanchon, and L. Trilling. Applications of a formal approach to decipher discrete genetic networks. *BMC Bioinformatics*, 11(1):385, 2010.

[16] M. W. Covert, C. H. Schilling, and B. Palsson. Regulation of gene expression in flux balance models of metabolism. *J. Theoret. Biol.*, 213(1):73 – 88, 2001.

[17] M. W. Covert, N. Xiao, T. J. Chen, and J. R. Karr. Integrating metabolic, transcriptional regulatory and signal transduction models in Escherichia coli. *Bioinformatics*, 24(18):2044–2050, 2008.

[18] M. Davidich and S. Bornholdt. The transition from differential equations to boolean networks: a case study in simplifying a regulatory network model. *J. Theoret. Biol.*, 255(3):269–277, 2008.

[19] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *J. Comput. Biol.*, 9:67, 2002.

[20] H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselmann. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull. Math. Biol.*, 66(2):301–340, 2004.

[21] H. de Jong, M. Page, C. Hernandez, and J. Geiselmann. Qualitative simulation of genetic regulatory networks: Method and application. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 67–73, 2001.

[22] H. de Jong and D. Ropers. Qualitative approaches towards the analysis of genetic regulatory networks. In Z. Szallasi, J. Stelling, and V. Periwal, editors, *System modeling in cellular biology*, pages 125–148. 2006.

[23] B. Derrida and Y. Pomeau. Random networks of automata: a simple annealed approximation. *Europhys. Lett.*, 1:45–49, 1986.

[24] E. Dimitrova, L. D. Garca-Puente, F. Hinkelmann, A. S. Jarrah, R. Laubenbacher, B. Stigler, M. Stillman, and P. Vera-Licona. Parameter estimation for boolean models of biological networks. *Theoret. Comput. Sci.*, 412(26):2816 – 2826, 2011.

[25] E. S. Dimitrova, M. P. Licona, J. McGee, and R. Laubenbacher. Discretization of time series data. *J. Comput. Biol.*, 17:853–868, 2010.

[26] M. Ederer, T. Sauter, E. Bullinger, E. Gilles, and F. Allgwer. An approach for dividing models of biological reaction networks into functional units. *Simulation*, 79(12):703 – 716, 2003.

[27] N. J. Eungdamrong and R. Iyengar. Computational approaches for modeling regulatory cellular networks. *Trends in Cell Biology*, 14(12):661 – 669, 2004.

[28] J. Fisher and T. A. Henzinger. Executable cell biology. *Nat. Biotech.*, 25(11):1230–1249, 2007.

[29] J. Fromentin, J.-P. Comet, P. Le Gall, and O. Roux. Analysing gene regulatory networks by both constraint programming and model-checking. In *Engineering in Medicine and Biology Society, EMBS 2007*, pages 4595 –4598, 2007.

[30] D. Gilbert, H. Fu, X. Gu, R. Orton, S. Robinson, V. Vyshemirsky, M. J. Kurth, C. S. Downes, and W. Dubitzky. Computational methodologies for modelling, analysis and simulation of signalling networks. *Brief. Bioinform.*, 7(4):339–353, 2006.

[31] L. Glass. Classification of biological networks by their qualitative dynamics. *J. Theoret. Biol.*, 54(1):85 – 107, 1975.

[32] L. Glass and S. A. Kauffman. The logical analysis of continuous, non-linear biochemical control networks. *J. Theoret. Biol.*, 39(1):103–129, 1973.

[33] L. Hartwell, J. Hopfield, S. Leibler, and A. Murray. From molecular to modular cell biology. *Nature*, 402:C47 – C52, 1999.

[34] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theoret. Biol.*, 22:437–467, 1969.

[35] S. A. Kauffman. *Origins of Order*. Oxford University Press, 1993.

[36] V. Kaufman, T. Mihaljev, and B. Drossel. Scaling in critical random boolean networks. *Phys. Rev. E*, 72(4):046124, 2005.

[37] H. Kitano. Systems biology: A brief overview. *Science*, 295(5560):1662–1664, 2002.

[38] R. Laubenbacher and P. Mendes. A discrete approach to top-down modeling of biochemical networks. In *Computational Systems Biology*, pages 229–247. Elsevier Academic Press, 2005.

[39] R. M. McConnell and F. de Montgolfier. Linear-time modular decomposition of directed graphs. *Discrete Appl. Math.*, 145(2):198 – 209, 2005.

[40] J. Min Lee, E. P. Gianchandani, J. A. Eddy, and J. A. Papin. Dynamic analysis of integrated signaling, metabolic, and regulatory networks. *PLoS Comput. Biol.*, 4(5):e1000086, 2008.

[41] P. Monteiro, D. Ropers, R. Mateescu, A. Freitas, and H. de Jong. Temporal logic patterns for querying dynamic models of cellular interaction networks. *Bioinformatics*, 24(16):i227–i233, 2008.

[42] A. Naldi, J. Carneiro, C. Chaouiya, and D. Thieffry. Diversity and plasticity of Th cell types predicted from regulatory network modelling. *PLoS Comput. Biol.*, 6(9):e1000912, 2010.

[43] A. Naldi, D. Thieffry, and C. Chaouiya. Decision diagrams for the representation and analysis of logical models of genetic networks. In *Computational Methods in Systems Biology, CMSB 2007, Edinburgh, Scotland*, volume 4695 of *LNBI*, pages 233–247, 2007.

[44] M. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci.*, 103(23):8577 – 8582, 2006.

[45] J. Reichardt and S. Bornholdt. When are networks truly modular? *Phys. D*, 224(1-2):20 – 26, 2006.

[46] E. Remy and P. Ruet. From minimal signed circuits to the dynamics of boolean regulatory networks. *Bioinformatics*, 24(16):i220–i226, 2008.

[47] E. Remy, P. Ruet, and D. Thieffry. Graphic requirements for multistability and attractive cycles in a boolean dynamical framework. *Adv. Appl. Math.*, 41(3):335–350, 2008.

[48] A. Richard. Negative circuits and sustained oscillations in asynchronous automata networks. *Adv. Appl. Math.*, 44(4):378 – 392, 2010.

[49] A. Richard and J.-P. Comet. Necessary conditions for multistationarity in discrete dynamical systems. *Discr. Appl. Math.*, 155(18):2403–2413, 2007.

[50] F. Robert. *Discrete Iterations: A Metric Study*, volume 6 of *Springer Series in Computational Mathematics*. Springer, 1986.

[51] J. Saez-Rodriguez, A. Kremling, and E. Gilles. Dissecting the puzzle of life: modularization of signal transduction networks. *Comput. Chem. Eng.*, 29(3):619 – 629, 2005.

[52] L. Sànchez, C. Chaouiya, and D. Thieffry. Segmenting the fly embryo: logical analysis of the role of the segment polarity cross-regulatory module. *Int. J. Dev. Biol.*, 52:1059–75, 2008.

[53] M. Schaub, T. Henzinger, and J. Fisher. Qualitative networks: a symbolic approach to analyze biological signaling networks. *BMC Syst. Biol.*, 1(1):4, 2007.

[54] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, 2002.

[55] I. Shmulevich and W. Zhang. Binary analysis and optimization-based normalization of gene expression data. *Bioinformatics*, 18(4):555–565, 2002.

[56] H. Siebert. Local structure and behavior of boolean bioregulatory networks. In *Algebraic Biology, AB 2008, Castle of Hagenberg, Austria*, volume 5147 of *LNCS*, pages 185–199, 2008.

[57] H. Siebert and A. Bockmayr. Context sensitivity in logical modeling with time delays. In *Computational Methods in Systems Biology, CMSB 2007, Edinburgh, Scotland*, volume 4695 of *LNBI*, pages 64–79, 2007.

[58] E. H. Snoussi. Qualitative dynamics of piecewise-linear differential equations: a discrete mapping approach. *Dyn. Syst.*, 4(3):565–583, 1989.

[59] E. H. Snoussi and R. Thomas. Logical identification of all steady states: the concept of feedback loop characteristic states. *Bull. Math. Biol.*, 55:973–991, 1993.

[60] C. Soulé. Mathematical approaches to differentiation and gene regulation. *Comptes Rendus Biologies*, 329(1):13 – 20, 2006.

[61] R. Thomas and R. d'Ari. *Biological Feedback*. CRC Press, 1990.

[62] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. ii. logical analysis of regulatory networks in terms of feedback circuits. *Chaos*, 11(1):180–195, 2001.

[63] L. Tournier and M. Chaves. Uncovering operational interactions in genetic networks using asynchronous boolean dynamics. *J. Theoret. Biol.*, 260(2):196–209, 2009.

[64] N. A. van Riel. Dynamic modelling and analysis of biochemical networks: mechanism-based models and model-based experiments. *Brief. Bioinform.*, 7(4):364–374, 2006.

[65] D. Wittmann, J. Krumsiek, J. Saez-Rodriguez, D. Lauffenburger, S. Klamt, and F. Theis. Transforming boolean models to continuous models: methodology and application to T-cell receptor signaling. *BMC Syst. Biol.*, 3(1):98, 2009.

# Part I

# Model Building

# Chapter 1

# Parameter Inference for Asynchronous Logical Networks Using Discrete Time Series

Hannes Klarner, Heike Siebert and Alexander Bockmayr.

In Proceedings of the 9th International Conference on *Computational Methods in Systems Biology*, CMSB 2011, Paris, France. To appear.

**Remarks.** The ideas and results presented in this paper were developed within the scope of Hannes Klarner's studies at Freie Universität Berlin. He is a doctoral student under the supervision of Alexander Bockmayr and myself. Prof. Bockmayr proposed the research topic of parameter inference for logic-based models using model checking techniques. Hannes Klarner developed the idea of encoding time series in computation tree logic to test the consistency of a model and given data. In an intensive supervision process, I discussed all ideas and developments with him, formalizing and significantly advancing the overall framework. In particular, I proposed the idea to not only focus on restricting the model pool using the time series data, but to also adapt the method to evaluate the sufficiency of measured data points. All authors worked jointly on the final version of the paper.

# Parameter Inference for Asynchronous Logical Networks Using Discrete Time Series

Hannes Klarner, Heike Siebert and Alexander Bockmayr

*Abstract.* This paper is concerned with the dynamics of asynchronous logical models of regulatory networks as introduced by R. Thomas. Available knowledge about the dynamics of a regulatory network is often limited to a sequence of snapshots in the form of a discrete time series. Using CTL formulas together with the concept of partially monotone paths, a methodology is elaborated to investigate the compatibility of a given time series and a Thomas model. The approach can be used to revise the model, but also to evaluate the given data. Additionally, suggestions are made to analyze a model pool for common properties regarding component behavior and interaction types, aiming at results exploitable for experimental design.

## 1 Introduction

In molecular biology, a regulatory network is a description of interactions between components. By assigning activity levels to the components and allowing interacting components to influence their activities depending on parameter values, such networks can be used to describe the system's dynamics in a state space. Since a full set of kinetic parameters is often not available, discrete modeling frameworks with finite parameter space have been suggested as an alternative to systems of differential equations.

Formal methods can help in determining suitable values for discrete parameters, translating available data into constraints on the set of all possible parameter choices, see e.g. Batt et al. [1] or Corblin, Fanchon, Trilling [7]. In this article, we employ similar ideas to test assumptions about component interplay for consistency. In case of inconsistencies, new hypotheses are systematically derived that then can be investigated experimentally. In contrast to related work, we additionally use our methods to evaluate the given experimental data by analyzing time series for potential ranges of poor sampling.

The paper is organized as follows. In Section 2 we recall the logical framework for regulatory networks and temporal logic. In Section 3 we introduce the notion of discrete time series as an ordered sequence of partial states. Section 4 elaborates a method of incorporating specific assumptions about monotonicity in between partial states. These are related to potential unobserved oscillations and can be used to evaluate the sufficiency of the provided data. In Section 5 we suggest a modeling workflow utilizing our methods, assessing the modeling assumptions as well as the quality of a given time series in terms of its temporal resolution, and discuss scalability and computational issues. We illustrate the procedure using an application example in Section 6, and conclude the paper discussing perspectives and future work.

# 2 Preliminaries

This section introduces our discrete modeling framework and model checking terminology. Throughout, discrete intervals will be denoted by

$$[a, b] := \{k \in \mathbb{N} \mid a \leq k \leq b\}, \quad \text{for } a, b \in \mathbb{N}.$$

The in- and out-degrees of a vertex of a graph are denoted by $d_-(v)$ and $d_+(v)$ and its predecessor and successor sets by $V_-(v)$ and $V_+(v)$, respectively.

## 2.1 Regulatory Networks

The discrete framework for modeling regulatory systems as introduced by Thomas in [15] consists of an edge-labeled digraph called regulatory network and a set of integer parameters.

**Definition 2.1** (Regulatory Network)**.** *A regulatory network $G = (V, E, t)$ is a directed graph with vertices $V := [1, n]$ for some fixed $n \in \mathbb{N}$, edges $E \subseteq V \times V$, maximal activity levels*

$$p : V \to [0, max(1, d_+(v))],$$

*and a function*

$$t : E \to \mathbb{N}, \text{ such that } t(u, v) \in [0, p(u)]$$

*that assigns thresholds to the edges $e \in E$. Nodes are called components and edges are called interactions. For a component $v \in V$, a predecessor $w \in V_-(v)$ is called a regulator of $v$ and a subset of regulators $R \subseteq V_-(v)$ is called a regulatory context of $v$.*

The vertices of the graph can be interpreted as variables taking values in the respective activity level interval $[0, p(v)]$. In the simplest case all variables are boolean. The edge labels are integers that represent thresholds above which regulatory interactions become effective.

**Definition 2.2** (Parameter Set)**.** *Given a regulatory network $(V, E, t)$, a parameter set $K = \{K_v \mid v \in V\}$ is a set of functions*

$$K_v : 2^{V_-(v)} \to [0, p(v)].$$

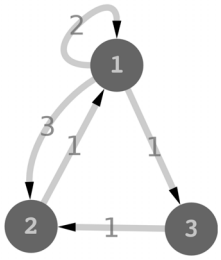
*$K_v$ is also called $v$-parameter subset of $K$.*

The network and parameter set in Fig. 1 will serve as a running example throughout the paper. Here we choose the maximal activity levels $p(v) = d_+(v)$ for all vertices. Any collection of parameter sets of a regulatory network is called a parameter pool. In particular, we define:

**Definition 2.3** (Parameter Space)**.** *The collection of all parameter sets of a regulatory network $(V, E, t)$ is denoted by*

$$\mathcal{K}(V, E, t) := \{K \mid K \text{ is a parameter set of } (V, E, t)\}$$

*and called the parameter space of $(V, E, t)$.*




| $R$ | $K_1(R)$ |
|---|---|
| $\emptyset$ | 0 |
| $\{1\}$ | 2 |
| $\{2\}$ | 3 |
| $\{1,2\}$ | 1 |

| $R$ | $K_2(R)$ |
|---|---|
| $\emptyset$ | 0 |
| $\{1\}$ | 0 |
| $\{3\}$ | 0 |
| $\{1,3\}$ | 1 |

| $R$ | $K_3(R)$ |
|---|---|
| $\emptyset$ | 0 |
| $\{1\}$ | 1 |

Figure 1: Example network and parameter set

The number of sets in the parameter space depends on the maximal activity levels of the network components and their in-degrees:

$$|\mathcal{K}(V,E,t)| = \prod_{v\in V}(p(v)+1)^{2^{d_-(v)}}.$$

The size of the parameter space of our running example is $|\mathcal{K}(V,E,t)| = 4^4 \cdot 2^2 \cdot 2^4 = 16384$.

The dynamics of a regulatory network $(V,E,t)$ with parameters $K$ is represented by a directed graph, called the state transition graph. It can be thought of as the discrete analogue to all possible trajectories in the phase plane of an ODE model. The nodes of this graph represent the discrete states of the system.

**Definition 2.4** (State Space)**.** *Given a regulatory network $(V,E,t)$, the state space $X$ is given by*

$$X = \prod_{v\in V}[0,p(v)].$$

To define the transitions between states it is convenient to turn the parameter set $K$ into a function $F$ on the state space $X$, where

$$F : X \to X, \quad x \mapsto F(x) = (f_1(x),\ldots,f_n(x)).$$

The image of $x$ under component function $f_v$ is defined to be a particular parameter $K_v(R)$. To choose this particular parameter we define the present regulators of $v$ in a state $x$.

**Definition 2.5** (Present Regulators)**.** *Given a regulatory network $(V,E,t)$ with parameters $K$ and its associated state space $X$, the present regulators $R_v(x)$ of a component $v \in V$ in state $x \in X$ are*

$$R_v(x) := \{w \in V \mid (w,v) \in E \wedge x_w \geq t(w,v)\}.$$

The present regulators of $v$ in state $x$ are components $w$ that regulate $v$ and whose activity level in state $x$ is above the threshold $t(w,v)$. This definition is the one given by Chaouiya et al. in [5]. With this notation the image of $x$ under $F$ is now defined to be

$$F(x) := (K_1(R_1(x)),\ldots,K_n(R_n(x))).$$

The present regulators of component 3 of the running example in state $x = (1,1,0)$ are $R_1(x) = \{2\}$, $R_2(x) = \emptyset$ and $R_3(x) = \{1\}$. Thus $F(x) = (2,0,1)$ according to the table given in Fig. 1.

There are several strategies for obtaining transitions using $F$. Most common are synchronous, asynchronous or priority strategies. The method described in this article is designed to work with unitary asynchronous transition rules. A simple notation for the transitions of the unitary asynchronous state transition graph is achieved with the tendencies $f'_v$ of the component functions $f_v$ of $F = (f_1, \ldots, f_n)$, as suggested by Richard in [13].

**Definition 2.6** (Tendencies)**.** *The tendency $f'$ of a component function $f_v : X \to \mathbb{N}$ is defined to be*

$$f'_v(x) = \begin{cases} 1: & f_v(x) - x_v > 0 \\ 0: & f_v(x) - x_v = 0 \\ -1: & f_v(x) - x_v < 0 \end{cases}.$$

The tendency of the second component in the state $(0, 1, 0)$ of the example parameter set is $f'_2(0, 1, 0) = -1$.

**Definition 2.7** (State Transition Graph)**.** *Given a regulatory network $(V, E, t)$ with parameter set $K$, the (unitary) asynchronous state transition graph is a directed graph $(X, T)$, where the nodes $X$ are the elements of the state space associated with the regulatory network and the edges $T$ are transitions between states. We have $(x, y) \in T$ iff either $y = x = F(x)$ or*

$$\exists v : f'_v(x) \neq 0 \wedge y = x + e_v f'_v(x_v),$$

*where $e_v$ is the $v$-th unit vector.*

The behavior represented in such a state transition graph is non-deterministic. In a given state $x$ there may be several $v \in V$ with $f'_v(x) \neq 0$, and therefore several $y$ with $(x, y) \in T$.

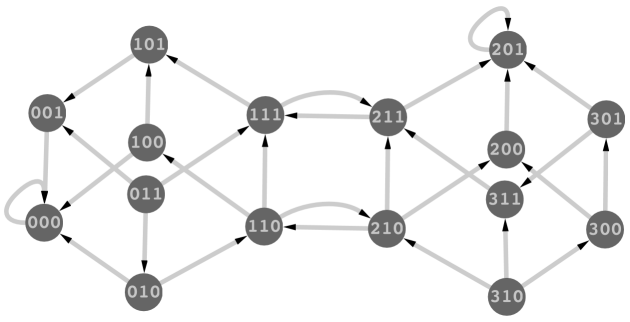



Figure 2: The state transition graph of the example parameter set.

## 2.2 Edge Constraints

Now, we consider information about the interaction type in the form of edge constraints. For example, interactions may be characterized as activating or inhibiting. Formally, edge constraints are additional edge labels that constrain the pool of feasible parameter sets for a regulatory network. Definitions for such edge constraints exist and are based on the observability or monotonicity of an interaction sign, usually either a '+' or a '-'. We use a more extensive set of edge

constraints to allow for a more precise characterization of individual interactions. Similar ideas can be found in [7].

This more general form of edge constraint is based on the observation that for a parameter set $K$, we can note for each interaction $(w, v)$ if there is a regulatory context $R \subseteq V_-(v)$, such that adding $w$ to $R$ increases or decreases the value of $K$ (as in [2]).

**Definition 2.8** (Increase and Decrease)**.** *Given a parameter subset $K$ of a regulatory network $(V, E, t)$, we define the boolean propositions $+$ and $-$ on the set of edges $(w, v) \in E$ by*

$$+(w, v) := \exists R \subseteq V_-(v) : K_v(R) < K_v(R \cup \{w\}),$$
$$-(w, v) := \exists R \subseteq V_-(v) : K_v(R) > K_v(R \cup \{w\}).$$

It has been remarked by Richard in the context of deriving global interaction graphs from dynamics (see [13]) that such a comparison of parameter values with and without a regulator $w$ is too weak to guarantee an effect observable in the state transition graph. For stronger results a slightly more technical definition of increase and decrease could be introduced here. For boolean networks and for components $v$ without self-regulation, i.e. $(v, v) \notin E$, the two definitions coincide.

For the parameter set of the running example, the values of $+$ and $-$ for each edge are the following:

$$\begin{array}{lll} +(1,1) = 1, & +(3,2) = 1, & +(1,3) = 1, \\ +(2,1) = 1, & +(1,2) = 1, & -(1,3) = 0. \\ -(1,1) = 1, & -(3,2) = 0, & \\ -(2,1) = 1, & -(1,2) = 0, & \end{array}$$

Instead of $\neg+$ and $\neg-$ we write $\overline{+}$ and $\overline{-}$. Simple logical expressions of these propositions are used to select parameter sets, by defining the following constraints.

**Definition 2.9** (Edge Constraints)**.**
*A labeling function*

$$s : E' \subseteq E \to \{+, \overline{+}, -, \overline{-}, + \wedge -, + \vee -, \overline{+} \wedge -, + \wedge \overline{-}\}$$

*on a subset $E' \subseteq E$ of the edge set of a regulatory network $(V, E, t)$ is called edge constraint. A parameter set $K$ satisfies the edge constraint $s$, if $s(w, v)$ is true for all $(w, v) \in E'$. In particular $\mathcal{K}(V, E, t, s)$ denotes all $K \in \mathcal{K}(V, E, t)$ that satisfy the edge constraint $s$.*

If an edge is not labeled by $s$, then no constraints are placed on the respective parameter values. The different labels can be interpreted as follows. $+$ and $-$ signify that an activating or inhibiting effect has been experimentally observed. It is not precluded that the respective opposite effect may also occur, depending on specific co-factors. In contrast, $+ \wedge \overline{-}$ and $\overline{+} \wedge -$ are used if the target is strictly activated or inhibited. $\overline{+}$ and $\overline{-}$ allow for the possibility that there is no interaction at all, but if so it is not activating respectively inhibiting. If the character of an interaction is not known or questionable but some effect is assumed, e.g., based on binding site properties, $+ \vee -$ is used. Finally, $+ \wedge -$ applies when the target is activated in some context and inhibited in another, reflecting the importance of co-factors.

Other logical combinations or types of edge constraints could be considered, for example, labeling the components by *max* or *min*:

$$max(v) := \exists R \subseteq V_-(v) : K_v(R) = d_+(v)\,,$$
$$min(v) := \exists R \subseteq V_-(v) : K_v(R) = 0\,.$$

However, the increase and decrease edge constraints already allow for a detailed description of interactions and suffice to illustrate the underlying method.

The parameter set pool $\mathcal{K}(V, E, t, s)$ can be efficiently computed with a backtracking algorithm.

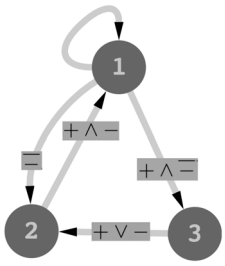



Figure 3: Edge constraints for example network, which lead to $|\mathcal{K}(V, E, t, s)| = 432$.

## 2.3 Model Checking

In this section, model checking is introduced as a means to analyze the state transition graph associated with a regulatory network. This has been proposed by various groups, see e.g. [2, 4, 9, 11, 7]. A Kripke structure or transition system is a state transition graph together with a labeling function that assigns atomic formulae to each node of the graph, which are defined to be true in this node. Computation Tree Logic (CTL) is a language that extends boolean propositions by temporal operators (see [10]). Boolean propositions can be evaluated at a node and so can CTL formulae. But, the temporal operators allow making statements about atoms that belong to other states, if there is a directed path in the transition graph from the first to the latter. Symbolic model checking is a fast method for finding the states in which a given CTL formula is true. We now will shortly review how to label the states of a transition graph, define the syntax of CTL and describe the semantics of CTL formulae.

A state transition graph $(X, T)$ can naturally be interpreted as a Kripke structure. Each state $x = (x_1, \ldots, x_n)$ has $n$ labels of the form "$v_i \doteq x_i$". Here, we write "$\doteq$" to distinguish syntactic from semantic equality. This labeling is extended to make the formula constructions in Section 4 possible.

**Definition 2.10** (State Transition System)**.** *Given a state transition graph $(X, T)$ with variables $V := \{v_i \mid i \in [1, n]\}$, the set of atomic formulas consists of equalities*

$$\mathcal{P} := \{ \sum_{1 \le i \le n} k_i v_i \doteq k \mid v_i \in V, k \in [-N, N], k_i \in \{-1, 0, 1\}\},$$

*where* $N := \sum_{v \in V} p(v)$. *Then* $(X, T, L)$, *with* $L : X \to 2^{\mathcal{P}}$ *and*

$$L(x) = \{ \sum_{1 \leq i \leq n} k_i v_i \doteq k \mid k_i \in \{-1, 0, 1\}, k = \sum_{1 \leq i \leq n} k_i x_i \},$$

*is the Kripke structure associated with the state transition graph* $(X, T)$.

A label $\sum_{1 \leq i \leq n} k_i v_i \doteq k$ captures simple expressions in the variables $v_i$ that are true in state $x$. Model checking software like NuSMV (see [6]) can handle such expressions. The number $N$ is included in the definition to emphasize that each node is only labeled with finitely many atoms. Here are a few atoms of the state $x = (0, 2, 11)$: $v_1 \doteq 0$, $v_1 + v_2 \doteq 2$, $-v_1 - v_2 + v_3 \doteq 9$.

The following definition of the syntax of CTL formulas is restricted to the temporal operators **EF** and **E**[ **U** ] that are needed for the method described here.

**Definition 2.11** (Syntax of CTL fragment)**.** *A CTL formula* $\phi$ *is defined inductively using the Backus Naur form. Let* $p$ *be an element of the set of atomic formulas* $\mathcal{P}$. *Then*

$$\phi ::= p \mid \phi \wedge \phi \mid \boldsymbol{EF}\phi \mid \boldsymbol{E}[\phi \ \boldsymbol{U} \ \phi].$$

Given a Kripke structure $(X, T, L)$, a state $x \in X$ and a CTL formula $\phi$, the following rules determine whether $\phi$ is true in $x$.

**Definition 2.12** (Semantics of CTL)**.**

- *An atomic formula* $p \in \mathcal{P}$ *is true in* $x$, *if* $p$ *is a label of* $x$, *i.e.,* $p \in L(x)$.
- $\phi \wedge \phi'$ *is true in* $x$, *if* $\phi$ *is true in* $x$ *and* $\phi'$ *is true in* $x$.
- $\boldsymbol{EF}\phi$ *is true in* $x$, *if* $\phi$ *is true in* $x$ *or if there is a path* $(x, x^1, \ldots, x^n)$ *in* $(X, T)$ *with* $n \geq 1$ *and* $\phi$ *is true in* $x^n$.
- $\boldsymbol{E}[\phi \ \boldsymbol{U} \ \phi']$ *is true in* $x$, *if* $\phi'$ *is true in* $x$ *or if there is a path* $(x, x^1, \ldots, x^n)$ *in* $(X, T)$ *with* $n \geq 1$ *such that* $\phi$ *is true in* $x$ *and* $x^i$ *for* $1 \leq i \leq n-1$ *and* $\phi'$ *is true in* $x^n$.

In the following sections, CTL formulas will be used to select parameter sets from given parameter pools. The selection is based on the existence of a state satisfying the formula.

**Definition 2.13** ($\phi$-Acceptable Parameter Sets)**.** *Given a CTL formula* $\phi$, *the collection of parameter sets of a regulatory network* $(V, E, t)$ *whose associated transition system contains a state in which* $\phi$ *is true is denoted by*

$$\mathcal{K}(V, E, t, \phi) := \{K \mid K \in \mathcal{K}(V, E, t) \wedge \exists x \in X : \phi \text{ is true in } x\}.$$

Sometimes a transition system is said to satisfy a CTL formula $\phi$, if $\phi$ is true in all states. Since we want to query the existence of paths starting in some state of the graph, the above definition is used.

## 3 Discrete Time Series

A discrete time series for a regulatory network can be obtained by discretizing real-valued experimental data or by qualitative observations about regulatory components. The issue of choosing a suitable discretization method for experimental data is crucial (see e.g. [8]), but is not the subject of this article. Under the assumption that the regulation behaves switch-like regarding the regulator concentration, one ideally has to estimate the threshold below which the regulator is not effective and above which it becomes effective.

If estimation is not possible, statistical approaches can be used, for example mean clustering, scan-statistic or edge gradient methods as described by Shmulevich and Zhang in [14]. There is also a software implementation for the GNU project R called BoolNet by Muessel et al. [12] which automates such discretization. BoolNet is used in Section 6 to discretize the expression data of the IRMA network [3].

Including qualitative observations in the time series is a strength of discrete modeling as it may be hard to translate such assumptions into quantitative data required for continuous models.

Mathematically, a discrete time series is a matrix where rows are measurements and columns are observations for one component. Data points with questionable discretization results for certain components or observations known to be imprecise may be recorded as uncertain by the value $-1$. In practice this has the advantage of deriving results based on varying levels of certainty.

**Definition 3.1** (Time Series)**.** *A discrete time series with $m$ measurements of $n$ substances is a matrix $A \in N^{m\times n}$, where the entries of $A$ are elements of $N := \mathbb{N} \cup \{-1\}$ and additionally*

$$\forall i \in [1,m] : \ \exists j \in [1,n] : \quad a_{i,j} \geq 0.$$

The condition ensures that measurements without supportable entries are not included in the time series.

As a discrete time series for the running example, including 4 measurements and 3 imprecise observations, we choose

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 2 & -1 & 1 \\ -1 & 1 & 0 \\ 3 & 0 & -1 \end{pmatrix}.$$

A time series will be interpreted as encoding discrete paths. To define these paths, the partial state formulas, one for each measurement, are derived. The definition uses the set of indices whose variables are not equal to $-1$. Thus, uncertain variables will be excluded from the description of the paths.

**Definition 3.2** (Partial States)**.** *Given a time series $A \in N^{m\times n}$, the partial state formula of measurement $i \in [1,m]$ is*

$$\sigma_i := \bigwedge_{j\in M_i} (v_j \doteq a_{i,j}), \qquad \textit{where} \qquad M_i := \{j \in [1,n] \mid a_{i,j} \geq 0\}.$$

A partial state formula may be true in a set of states, depending on how many variables are uncertain. The paths encoded in a time series are then all paths that connect the partial states in the given order. A state transition graph that contains at least one such path is said to be able to reproduce the time series.

**Definition 3.3** (Reproducing a Time Series)**.** *A state transition graph $(X, T)$ can reproduce a time series $A = (a_{i,j}) \in N^{m \times n}$ if there is a path $(x^1, \ldots, x^k)$ in $(X, T)$ such that the index sequence $(1, \ldots, k)$ has a subsequence $(r_1, \ldots, r_m)$ satisfying for each $1 \leq i \leq m$ that $\sigma_i$ is true in $x^{r_i}$.*

*We say a parameter set can reproduce a time series, if this holds for the corresponding state transition graph.*

The sequence of states $(x^1, \ldots, x^k)$ can be thought of as a simulation of the regulatory network from the initial state $x^1$. An intuitive CTL formula can be used to check if a parameter set can reproduce a time series. Such a formula is a nested sequence of partial state formulas connected via the predicates **EF** :

$$\sigma_1 \wedge \mathbf{EF}[\sigma_2 \wedge \mathbf{EF}[\ldots \sigma_{m-1} \wedge \mathbf{EF}[\sigma_m] \ldots]].$$

# 4 The Monotone Path Formulas

In this section, the paths encoded in a time series are characterized with regard to monotonicity in between successive measurements. The motivation for this is to take into account assumptions about the ratio of time elapsed between measurements on the one hand, and rates of change of components on the other. Intuitively, if for a substance the time elapsed between successive measurements is small compared to its rate of change, then we would expect its concentration to change monotonously, i.e., without oscillations.

To encode these ratios for each variable and at each measurement, we define a matrix to specify exactly which parts of the path should be monotone.

**Definition 4.1** (Monotonicity Matrix)**.** *Given a discrete time series $A \in N^{m \times n}$, a monotonicity matrix of $A$ is any matrix $B = (b_{i,j}) \in \{0, 1\}^{m-1,n}$ such that*

$$\forall i, j: \quad b_{i,j} = 1 \implies (a_{i,j} \geq 0 \wedge a_{i+1,j} \geq 0).$$

*We say that variable $j$ is specified to be monotone at measurement $i$, iff $b_{i,j} = 1$.*

A time series and a monotonicity matrix define the following partially monotone paths. For technical reasons regarding the CTL construction in 4.5, we require that the path begins in a state representing the first and ends in one representing the last measurement.

**Definition 4.2** ($A$-$B$-Monotone Paths)**.** *Given a discrete time series $A \in N^{m \times n}$ together with a monotonicity matrix $B$, and a state transition graph $(X, T)$, a path $(x^1, ..., x^r)$ in $(X, T)$ is $A$-$B$-monotone, if there is a subsequence $(r_1, ..., r_m)$ of $(1, ..., r)$ with $r_1 = 1, r_m = r$ and the following two properties hold. First*

$$0 \leq a_{i,j} \implies x_j^{r_i} = a_{i,j}.$$

*Second, for the variables $j$ specified to be monotone at measurement $i$*

$$\forall t \in [r_i, r_{i+1} - 1] : \begin{cases} x_j^t \leq x_j^{t+1} : & \text{if } x_j^{r_i} \leq x_j^{r_{i+1}} \\ x_j^t \geq x_j^{t+1} : & \text{if } x_j^{r_i} > x_j^{r_{i+1}} \end{cases}.$$

A monotonicity matrix for the example time series $A$ is

$$B = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix},$$

and an example of an $A$-$B$-monotone path is

$$((0,1,0),(1,1,0),(1,1,1),(2,1,1),(2,1,0),(3,1,0),(3,0,0)).$$

Again, a CTL formula is constructed to check the existence of an $A$-$B$-monotone path in a transition system. This formula is specifically designed for asynchronous transition graphs. It exploits the observation that for each couple of successive measurements, there is an expression $\sum k_i v_i$ in the marked monotone variables $v_i$ that is increasing along any $A$-$B$-monotone path. To determine this expression we need to consider the variables that increase and decrease separately.

**Definition 4.3** (Index Sets)**.** *Given a discrete time series $A \in N^{m \times n}$ and a monotonicity matrix $B \in \{0,1\}^{m-1,n}$, we define for each $i \in [1, m-1]$ the index sets $M_i^+$ and $M_i^-$ of increasing and decreasing variables respectively:*

$$\begin{aligned} M_i^+ &:= \{j \in [1,n] \mid b_{i,j} = 1 \wedge\ a_{i,j} \leq a_{i+1,j}\}, \\ M_i^- &:= \{j \in [1,n] \mid b_{i,j} = 1 \wedge\ a_{i,j} > a_{i+1,j}\}. \end{aligned}$$

Now we can construct the increasing expression mentioned before, define its initial value and by how much it has to increase in between measurements.

**Definition 4.4** (Increasing Expression)**.** *The increasing expression $V_i = V_i(v_1, \ldots, v_n)$, the initial value $C_i$ and the distance $d_i$ for $i \in [1, m-1]$ are defined to be*

$$\begin{aligned} V_i &:= \sum_{j \in M_i^+} v_j + \sum_{j \in M_i^-} (a_{i,j} - v_j), \\ C_i &:= \sum_{j \in M_i^+} a_{i,j}, \\ d_i &:= \sum_{j \in M_i^+ \cup M_i^-} |a_{i,j} - a_{i+1,j}|. \end{aligned}$$

In a state satisfying the partial state formula $\sigma_i$, the atomic formula $V_i \doteq C_i$ is true. The following $A$-$B$-monotone path formula asserts that $V_i \doteq C_i$ increases one by one until $V_i \doteq C_i + d_i$ and $\sigma_{i+1}$ are true. To deal with the nested structure of the formula, it is defined recursively.

**Definition 4.5** ($A$-$B$-Monotone Path Formula)**.** *The $A$-$B$-monotone path formula $\phi_{A,B}$ for a time series $A \in N^{m \times n}$ and monotonicity matrix $B$ is constructed recursively using the formulae $\rho_i, i \in [1, m]$. Let*

$$\rho_1 := \sigma_m,$$

*and for* $i \in [1, m-1]$

$$\rho_{i+1} := \begin{cases} \sigma_{m-i} \wedge \boldsymbol{EF}[\rho_i] & \text{if } M^+_{m-i} \cup M^-_{m-i} = \emptyset \\ \sigma_{m-i} \wedge \gamma^{m-i}_{d_{m-i}+1} & \text{if } M^+_{m-i} \cup M^-_{m-i} \neq \emptyset. \end{cases}$$

*Here*

$$\gamma^{m-i}_1 := \boldsymbol{E}[(V_{m-i} \doteq C_{m-i} + d_{m-i}) \ \boldsymbol{U} \ \rho_i]$$

*and if* $d_{m-i} \geq 1$ *then*

$$\gamma^{m-i}_{t+1} := \boldsymbol{E}[(V_{m-i} \doteq C_{m-i} + d_{m-i} - t) \ \boldsymbol{U} \ \gamma^{m-i}_t]$$

*for* $t \in [1, d_{m-i}]$. *Finally, define* $\phi_{A,B} := \rho_m$.

From the above definition, a pseudo code algorithm for the construction of an $A$-$B$-monotone path formula is derived:

$\rho_1 := \sigma_m$
**for** $i = 1$ **to** $m-1$ **do**
  **if** $M^+_{m-i} \cup M^-_{m-i} = \emptyset$ **then**
    $\rho_{i+1} := \sigma_{m-i} \wedge \mathbf{EF}[\rho_i]$
  **else**
    $\gamma^{m-i}_1 := \mathbf{E}[(V_{m-i} \doteq C_{m-i} + d_{m-i}) \ \mathbf{U} \ \rho_i]$
    **if** $d_{m-i} \geq 1$ **then**
      **for** $t = 1$ **to** $d_{m-i}$ **do**
        $\gamma^{m-i}_{t+1} := \mathbf{E}[(V_{m-i} \doteq C_{m-i} + d_{m-i} - t) \ \mathbf{U} \ \gamma^{m-i}_t]$
      **end for**
    **end if**
    $\rho_{i+1} := \sigma_{m-i} \wedge \gamma^{m-i}_{d_{m-i}+1}$
  **end if**
**end for**

Next we show that this formula characterizes the existence of an $A$-$B$-monotone path.

**Theorem 4.6** (Correctness)**.** *Given an asynchronous state transition graph* $(X, T)$*, its associated state transition system* $(X, T, L)$ *and a discrete time series* $A \in N^{m \times n}$ *together with a monotonicity matrix* $B \in \{0, 1\}^{m-1,n}$*, the* $A$-$B$*-monotone path formula is true in* $(X, T, L)$ *if and only if there is an* $A$-$B$*-monotone path in* $(X, T)$.

*Proof.* By the recursive structure of $\rho_m$ it is sufficient to consider a matrix $A$ with just two rows. For further simplicity assume there are only increasing variables ($M^-_1 = \emptyset$). The mixed case follows the same reasoning, because every $j \in M^-_1$ appears as $v'_j := x^1_j - v_j$ in $V_1$ and $v'_j$ increases, if $v_j$ decreases.

First, we want to show that the existence of an $A$-$B$-monotone path $(x^1, ..., x^r)$ in $(X, T)$ implies that $\rho_2$ is true in $x^1$. For each $t \in [1, r-1]$ we have $V_1(x^t) \leq V_1(x^{t+1})$, because $V_1$ is the sum of variables that increase along that path. The difference $V_1(x^{t+1}) - V_1(x^t)$ is at most 1 since $T$ contains only unitary asynchronous transitions. So there must be a partition of $[1, r]$ into $d_1 + 1$ intervals, where $d_1 := \sum_{j \in M^+_1} (x^r_j - x^1_j)$, such that $V_1$ is constant on each interval and increases by 1 from one interval to the next. On all states $x$ of the first interval the formula

$V_1(x) \doteq C_1$ is true and on all states $x$ of the last interval the formula $V_1(x) \doteq C_1 + d_1$ is true. Therefore $\gamma_t^1$ for $t \in [1, d_1 + 1]$ is true on the $t$-th interval, counted from right to left and hence $\rho_2$ is true in $x^1$.

Second, we want to show that $\rho_2$ is true in $x \in X$ implies that there is an $A$-$B$-monotone path in $(X, T)$. Since $\rho_2$ is true in $x^1 := x$ there is a path $(x^1, ..., x^r)$ in $(X, T)$ such that $\sigma_1$ is true in $x^1$ and $\rho_1 \doteq \sigma_2$ is true in $x^r$, which is the first property of an $A$-$B$-monotone path. Furthermore, $[1, r]$ can be partitioned into $d_1 + 1$ intervals such that $\gamma_t^1$ is true in the $t$-th interval counted from right to left. Therefore $V_1$ increases by 1 from one interval to the next. Since $T$ contains only unitary asynchronous transitions, there is exactly one variable $j \in M_1^+$ that increases by 1 from one interval to the next. Therefore $x_j^k \leq x_j^{k+1}$ for all $k \in [1, r-1]$ and $j \in M_1^+$ which is the second property of an $A$-$B$-monotone path. So the path $(x^1, ..., x^r)$ is $A$-$B$-monotone. □

# 5 Workflow

In this section, we introduce a methodology to analyze compatibility of a regulatory network and a given time series. After describing a possible workflow, we conclude the section with remarks regarding computation methods and costs. The procedures are illustrated in the next section.

Let us consider a regulatory network, possibly including edge constraints, a time series and a monotonicity matrix (consisting only of zero entries in case no monotonicity assumptions are made). As a first step, we check whether there are parameter sets that reproduce the time series, i.e., we compute the parameter pool $\mathcal{K}(V, E, t, s, \phi_{A,B})$. If the model checking procedure returns a unique parameter set, we can proceed with the analysis of the model. However, this case will only occur very rarely. More commonly, the procedure either returns a large pool of parameter sets or no set at all. In the following, we look at both cases more closely

## 5.1 Characterizing Model Pools

If the parameter pool contains many parameter sets, the information encoded in the network and the time series was not sufficient to determine a unique specified model. One possibility to deal with this difficulty is to choose a model from the pool using meaningful criteria, e.g., some notion of minimality. A different approach is to characterize the parameter pool in order to derive information about the system strongly supported by the integrated data. We propose ideas in line with the second approach. One characteristic of a model pool are parameter values that are identical across all parameter sets. Such values may allow for new insights into how a component behaves under the influence of several regulators, clarifying synergies and redundancies in the network.

**Definition 5.1** (Determined Parameter Values)**.** *Given a parameter pool $\mathcal{K}$, the value of a component $v$ in a regulatory context $R \subseteq V_-(v)$ is determined if there is a $p \in [0, p(v)]$ such that*

$$\forall K \in \mathcal{K} : K_v(R) = p.$$

This idea can be extended to finding the range of values for each component and regulatory context.

Even if the parameters for a given component are not completely determined, we can still try to extract further information. To get an idea about the different behaviors that a component can have in a parameter pool, we count the $v$-local parameter sets in $\mathcal{K}$.

**Definition 5.2** (Behaviors)**.** *Given a parameter pool $\mathcal{K}$ of a regulatory network $(V, E, t)$, the behaviors $\mathcal{K}_v$ of component $v \in V$ are the set of $v$-local parameter sets in $\mathcal{K}$,*

$$\mathcal{K}_v := \{K_v \mid K \in \mathcal{K}\}.$$

This information can be used to study how components are tuned to work together in reproducing a time series. If any combination of component behaviors is a parameter set in the pool, then the components are said to be independent.

**Definition 5.3** (Independence)**.** *A parameter pool $\mathcal{K}$ consists of independent components, if*

$$\prod_{v \in V} |\mathcal{K}_v| = |\mathcal{K}|.$$

Further characterization of the parameter pool could study which behaviors do not appear together and try to identify components and regulatory contexts, which, if determined, would lead to the steepest reduction in feasible parameter sets. Identifying such contexts could be used to design experiments that reduce the number of feasible parameter sets in the fastest possible way.

Characterization of the parameter pool can also focus on the edge labels. They can be arranged into a logical implication hierarchy. For example. "$+ \wedge \overline{-} \implies \overline{-}$" and we thus place $+ \wedge \overline{-}$ above $\overline{-}$ in the hierarchy diagram in Fig. 4. For each unlabeled edge of the regulatory network and edges carrying one of the constraints that may be strengthened ($+$,$-$,$+ \vee -$,$\overline{-}$,$\overline{+}$), we determine the strictest label that is true for all parameter sets. This may lead to determining an effect of a regulator on its target that was formerly not known. An edge may for example be included in a network, because the source component is known to bind to the target component's promoter, but without any knowledge of the effect this binding has (i.e., with label $+ \vee -$). With a time series this label may be sharpened to $+$ and thus hypothesize an activation.

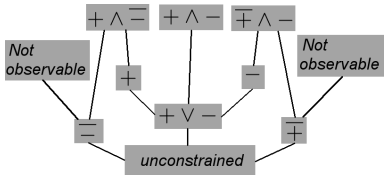



Figure 4: Hierarchy of edge constraints with stricter labels above weaker ones. "Not observable" is used to emphasize the meaning of $\overline{+} \wedge \overline{-}$.

## 5.2 Evaluating the Time Series

An ideal sampling frequency would result in a discrete time series capturing all value changes of the components, but usually data points are rather sparse. In order to understand the underlying system, we need to know whether the sampling was sufficient to capture its essential behavior. Here, we focus on determining potential oscillatory behavior not inferable from the time series due to coarse sampling.

Consider a network, a time series and a monotonicity matrix that are compatible, i.e., the corresponding parameter pool is not empty. We start with the assumption that the time series is sufficient to exclude the possibility of undetected oscillatory behavior. Intuitively, if sufficiently many measurements were made, it can be assumed that all variables are monotone at all measurements.

**Definition 5.4** (Best Fit)**.** *Given a regulatory network $(V, E, t)$, a time series $A$ and the monotonicity matrix $B$, where*

$$b_{i,j} = \begin{cases} 1: & 0 \leq a_{i,j}, a_{i+1,j} \\ 0: & else \end{cases},$$

*a parameter set that satisfies the A-B-monotone path formula is called a best fit of $(V, E, t)$ to $A$.*

Recall that the entries or positions $(i, j)$ of $B$ represent the value transition of the $j$-th component from measurement $i$ to measurement $(i+1)$, and that the entry 1 signifies a monotone value change. If no best fits of $(V, E, t)$ to $A$ exist, we can be sure that there is a set of positions of $B$, such that all parameter sets in the considered pool produce at least one unobserved oscillation in one of the positions. In these positions the temporal resolution of $A$ is too coarse to capture the behavior of the network. A trivial such set is the set of all positions, but there may be a smaller set, ideally with only a single position. Starting with the originally considered monotonicity matrix $B$, a heuristic approach to finding a non-trivial set is to introduce additional monotonicity constraints position by position. If such an added constraint does not result in a reduction of the parameter pool, we discard the corresponding position, since all models agree with the assumed monotonicity for that position, and we need no extra sampling between the corresponding data points. We introduce a measure for the impact of an additional monotonicity constraint as follows.

**Definition 5.5** (Selectivity)**.** *Given a regulatory network $(V, E, t)$, a time series $A$ and a monotonicity matrix $B$, we define for each $1 \leq i \leq m, 1 \leq j \leq n$ with $b_{i,j} = 0$ and $0 \leq a_{i,j}, a_{i+1,j}$, the monotonicity matrix $B'$ by*

$$b'_{i',j'} := \begin{cases} 1: & i' = i, j' = j \\ b_{i,j}: & else \end{cases}$$

*and the selectivity of position $(i, j)$ by*

$$S(i, j) := 1 - \frac{|\mathcal{K}(V, E, t, \phi_{A,B'})|}{|\mathcal{K}(V, E, t, \phi_{A,B})|}.$$

All positions that have selectivity 1 hypothesize obligatory oscillations of component $j$ in between measurements $i$ and $i + 1$, which indicates the need for additional data points between the measurements. If no such positions exist, we choose the set $\{(i, j) \mid S(i, j) > 0\}$ as places of interest for new measurements.

## 5.3 Reviewing Structure and Data

So far we have considered the case that we have no contradictions in our modeling assumptions and data, resulting in viable choices of parameter sets. If a network is not compatible with a

time series and the possibly additionally provided monotonicity matrix, i.e., the corresponding parameter pool is empty, there are two possible lines of investigation, depending on whether the correctness of the network structure or of the data is questioned. In both cases, the idea is to check what minimal changes can lead to compatibility.

Regarding the structure, we may, in a first step, relax the constraints on the interactions and instead label every edge with the observability label $+ \vee -$. Thus we include no assumptions on the character of an interaction, but only require it to be observable. We now test if the weakened assumptions result in a non-empty parameter pool.

**Definition 5.6** (Structural Compatibility)**.** *A regulatory network $(V, E, t)$ is called structurally compatible with a time series $A$ and a monotonicity matrix $B$, if, for the labeling function $s : E \to \{+ \vee -\}$, there exists a parameter set $K \in \mathcal{K}(V, E, t, s, \phi_{A,B})$ that can reproduce the time series.*

If the network is structurally compatible, we know that the contradiction must be caused by the assumed edge constraints and so a review of the edge labels (see 5.1) of the compatibility pool should reveal that there is an interaction whose label has changed. If it is not, we may go further and introduce additional edges or remove existing ones.

Regarding the data, we can proceed similarly by first lifting monotonicity constraints in $B$ (if there are any) and then replacing particular values in $A$ with the imprecise $-1$.

## 5.4 Computation and Scalability

The computational steps in the workflow are (1) to exhaustively generate all parameter sets satisfying the edge constraints (not the whole parameter space), (2) to translate a parameter set into a model checker input file, and (3) to pass it to a model checker, together with the $A$-$B$-monotone path formula. For model checking we use NuSMV ([6], see also [2] and [1]). For computation of the parameter sets, we apply a backtracking algorithm with failure on constraint violation.

Regarding scalability and computation times, we first note that the state space is exponential in the number of components, which places a strong limit on the possible number. Second, we compute a large part of the parameter space, depending on how restricting the edge labels are. Efficient algorithms considering partial parameterizations only have been introduced for PADE models (see [1]). Similar approaches would be desirable for the Thomas formalism.

As standing, analysis is limited to structures of about 30,000 states, e.g. 15 binary components or 9 ternary components. For such models the time per model check is impacted considerably by the nesting depth of a given CTL formula, which in our case increases linearly with the length of a time series and monotonicity constraints. Model checking a 30,000 state model and a time series of 5 measurements takes about 1 second on a 2.27GHz Laptop.

Given these restrictions imposed by the time per model check, the computation time for the parameter sets is negligible. However, it should be noted that even with the most restrictive edge labels ($+ \wedge \overline{-}$ and $- \wedge \overline{+}$) on edges targeting a binary component, there are already 6,894 local parameter sets for only 5 regulators. For a ternary component, the number of such regulators is limited to 4, resulting in 7,008 local parameter sets.

## 6 Application: The IRMA Network

We apply the workflow of the previous section to a biological network called IRMA, for which several time series are available. A corresponding search for consistent parameters of a qualitative PADE model is described by Batt et al. [1].

The IRMA regulatory network consists of 5 genes with gene control and protein-protein interactions, which has been inserted into the genome of Saccharomyces cerevisiae (see Cantone 2009 [3]).

Several populations of this genetically modified yeast were grown and subjected to perturbations by adding or removing galactose from the growth medium. Altogether 11 real-valued time series are available: 5 repetitions of the switch-on perturbation (adding galactose) and 4 repetitions of the switch-off perturbation (removing galactose) plus two averaged time series for each category.

A comprehensive analysis would include all available time series. Since we aim for a clear illustration of our approach, we restrict analysis to the averaged switch-off time series. In addition, we only consider a boolean model.

We binarized the expression data for the galactose removal experiment using the scan-statistic method described in [12]. Additionally, we added values for *gal* based on qualitative observations. The first entry of its profile is left uncertain, because although the cells were washed, we are not sure if galactose was still present in the cytoplasm or not. This resulted in the discrete time series

$$A = \begin{pmatrix} CBF1 & ASH1 & GAL4 & GAL80 & SWI5 & gal \\ 1 & 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Matching the binarized data, we chose a boolean representation, i.e., $p(v) = 1$ for every variable $v$. The state space is then

$$X = [0,1]^6 \text{ with } |X| = 64, \text{ and}$$

$$|\mathcal{K}(V,E,t)| = 2^2 \cdot 2^2 \cdot 2^4 \cdot 2^2 \cdot 2^2 \cdot 2^8 = 2^{20} = 1.048.576.$$

The network edges and edge-constraints, as shown in Fig. 5, were adopted from [3] and interpreted as '+' and '−', i.e., as observable activations or inhibitions. We then computed all parameter sets that satisfy the edge-constraints and reproduce the time series without any

monotonicity assumptions:

$$|\mathcal{K}(V, E, t, s)| = 404 \text{ and } |\mathcal{K}(V, E, t, s, \phi_{A,0})| = 73.$$

We proceeded by characterizing the parameter pool $\mathcal{K}(V, E, t, s, \phi_{A,0})$. All parameters of components with a unique regulator, namely $GAL4$, $ASH1$ and $GAL80$, coincide for all parameters sets, i.e., the component behavior is completely determined. The labels of edges targeting these components can be strengthened to $+ \wedge \overline{-}$, i.e., they are recognized as non-ambiguous activating influences. For $SWI5$ one parameter is determined: $K_{SWI5}(\{GAL4\}) = 1$, suggesting that $GAL4$ alone is sufficient to activate $SWI5$, as opposed to galactose which may require $GAL4$ for up-regulation of $SWI5$ as the parameter $K_{SWI5}(\{gal\})$ is in the range $[0, 1]$.

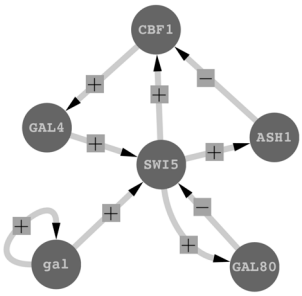



Figure 5: The IRMA regulatory network.

Regarding the behaviors of $CBF1$ and $SWI5$ as defined in Sect. 5.1, there are 4 for the latter and 33 for the former. The set $\mathcal{K}(V, E, t, s, \phi_{A,0})$ is not independent, since $4 \cdot 33 = 132$, but there are only 73 sets in the pool. Therefore, not every behavior of $SWI5$ is compatible with every behavior of $CBF1$. Identification of conflicting behaviors can then be utilized for experimental design. Development of strategies that allow to identify a component and corresponding behavior whose parameter determination would result in a maximal decrease of the parameter pool is an issue for future work.

Continuing in the workflow, we assessed the quality of the time series. There are no best fits of the IRMA network to the time series, but computing the selectivity of positions $(i, j)$ in $A$ we found 8 positions to have a selectivity of 1 and hypothesize the following oscillations.

| Name | Begins oscillation at measurement |
|---|---|
| $CBF1$ | 1,8,11 |
| $SWI5$ | 5,7,8,11,12 |

The real-valued expression profiles show that $SWI5$ does indeed oscillate, but that the oscillations are below the threshold that the binarization method computed. In this particular case, the result emphasizes the need of revising the chosen threshold. However, it also illustrates nicely the potential of our method to evaluate sufficiency of measurements, since similar results would be obtained if the data points between 5 and 15 in the $SWI5$ plot were simply missing. Based on our analysis the importance of providing additional measurements for that time span would be highlighted.

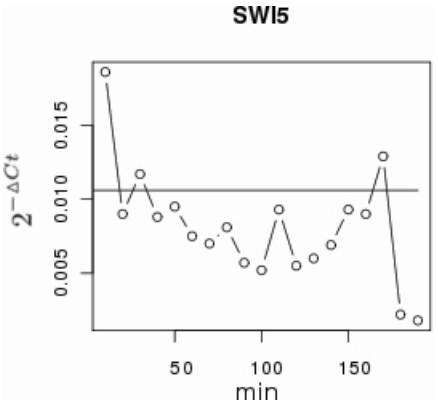



Figure 6: Real-valued expression profile of $SWI5$ where the horizontal line is the binarization threshold obtained by the scan-statistic method.

For $CBF1$ the expression curve shows a decline with two steady intervals around measurements 10 and 15. Here, the real-valued data shows no oscillation, but rather different plateaus. Our results point out the time points where changes of activity levels result in qualitatively observable effects, and thus indicate the need for a finer representation of activity levels than a simple boolean view. Investigating the relation between the predictions for oscillations generated by our method and the need for an expanded component value range will be an objective of future work.

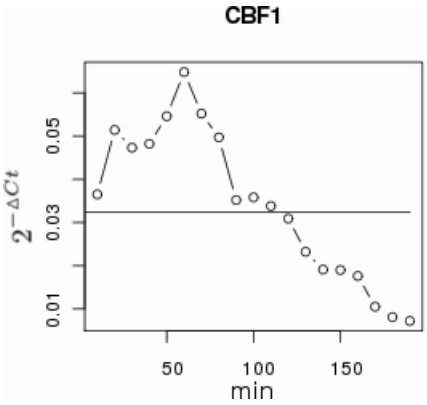



Figure 7: Real-valued expression profile of $CBF1$ where the horizontal line is the binarization threshold obtained by the scan-statistic method.

Since the boolean model for the IRMA network can reproduce the chosen time series, we imposed additional assumptions to illustrate the workflow in case of inconsistencies (Sect. 5.3). We considered that value changes in $GAL80$ involve transcription processes. Let us assume that the transcription of $GAL80$ is slow, so that it is not expected to significantly change concentration within the sampling rate of 10 minutes, i.e., there will be no oscillations between the sampling points.

The entries of a monotonicity matrix $B$ encoding this assumption are 1 in the column corresponding to the $GAL80$ expression profile. We set all remaining entries of $B$ to zero, imposing no further monotonicity constraints. The corresponding parameter pool $\mathcal{K}(V, E, t, s, \phi_{A,B})$ is empty. We decided to proceed by revising the structure of the internal components, taking the activating effect of *gal* on $SWI5$ as given. The IRMA network is structurally compatible with $A$ and $B$. We now try to derive valid information from the resulting parameter pool. Of the 12,960 parameter sets in the pool where all internal edges of the network, i.e. not $(gal, gal)$ and $(gal, SWI5)$, are relaxed to $+ \vee -$, 144 satisfy $\phi_{A,B}$. Interestingly there are no determined parameter values in this pool, but two interactions are stricter than assumed in every parameter set:

$$s(ASH1, CBF1) = -, \quad s(SWI5, CBF1) = +.$$

This illustrates how we can recover information from the parameter pool supported by the available data. In summary, we can observe that the reasonable assumption that the switch-off series has captured all oscillations of $GAL80$ validates the original labels targeting $CBF1$.

# 7 Conclusion

In this paper we study the compatibility of a model of a regulatory network and its observed behavior in the form of a discretized time series. On the formal level, we slightly extend the usual edge labels (e.g. [2]) with boolean propositions on edges (similar to [7]) and introduce time series that may be partially exact or monotone. On the methodological level, a workflow is suggested that branches in places where given assumptions may or may not be satisfied.

In contrast to related work, we also use our methods to assess the quality of the considered time series. In case of consistency of the network structure and the time series, we investigate the temporal resolution of the time series by defining a best fit. For such parameter sets additional measurements would not reveal much further information, because in between measurements all variables approach their target activities without oscillating. However, we show that if no best fits exist, oscillations can be predicted for particular variables in particular time intervals. We have shown the potential of this approach using the IRMA network. In addition, the results hint at the possibility of using the same methods to assess the discretization threshold of individual components, as well as the number of thresholds used for a component. This will be further elucidated in future work.

While we obtain satisfactory results for networks of small and medium size, we certainly have to increase the computation efficiency to tackle larger models. Future research will focus on development of more powerful implementations of our ideas.

# References

[1] G. Batt, M. Page, I. Cantone, G. Goessler, P. Monteiro, and H. de Jong. Efficient parameter search for qualitative models of regulatory networks using symbolic model checking. *Bioinformatics*, 26(18):i603–i610, 2010.

[2] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. Application of formal methods to biological regulatory networks: extending Thomas' asynchronous logical approach with temporal logic. *J. Theoret. Biol.*, 229(3):339 – 347, 2004.

[3] I. Cantone, L. Marucci, F. Iorio, M. A. Ricci, V. Belcastro, M. Bansal, S. Santini, M. di Bernardo, D. di Bernardo, and M. P. Cosma. A yeast synthetic network for in vivo assessment of reverse-engeneering and modeling approaches. *Cell*, 137(1):172–181, 2009.

[4] N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter. Modeling and querying biomolecular interaction networks. *Theoret. Comput. Sci.*, 325(1):25 – 44, 2004.

[5] C. Chaouiya, E. Remy, B. Mossé, and D. Thieffry. Qualitative analysis of regulatory graphs: A computational tool based on a discrete formal framework. In L. Benvenuti, A. De Santis, and L. Farina, editors, *Positive Systems*, volume 294 of *Lecture Notes in Control and Information Sciences*, pages 830–832. 2003.

[6] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. Nusmv 2: An opensource tool for symbolic model checking. In E. Brinksma and K. Larsen, editors, *Computer Aided Verification*, volume 2404 of *LNCS*, pages 241–268. 2002.

[7] F. Corblin, E. Fanchon, and L. Trilling. Applications of a formal approach to decipher discrete genetic networks. *BMC Bioinformatics*, 11(1), 2010.

[8] E. S. Dimitrova, M. P. Licona, J. McGee, and R. Laubenbacher. Discretization of time series data. *J. Comput. Biol.*, 17:853–868, 2010.

[9] J. Fromentin, J.-P. Comet, P. Le Gall, and O. Roux. Analysing gene regulatory networks by both constraint programming and model-checking. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007*, pages 4595 –4598, 2007.

[10] M. Huth and M. Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems*. Cambridge University Press, 2000.

[11] P. Monteiro, D. Ropers, R. Mateescu, A. Freitas, and H. de Jong. Temporal logic patterns for querying dynamic models of cellular interaction networks. *Bioinformatics*, 24(16):i227–i233, 2008.

[12] C. Muessel, M. Hopfensitz, D. Zhou, and H. Kestler. R package 'boolnet', 2010. CRAN repository: http://cran.r-project.org/web/packages/BoolNet.

[13] A. Richard. Negative circuits and sustained oscillations in asynchronous automata networks. *Adv. in Appl. Math.*, 44(4):378 – 392, 2010.

[14] I. Shmulevich and W. Zhang. Binary analysis and optimization-based normalization of gene expression data. *Bioinformatics*, 18(4):555–565, 2002.

[15] R. Thomas. Regulatory networks seen as asynchronous automata: a logical description. *J. Theoret. Biol.*, 153(1):1–23, 1991.

# Part II

# Analysis

# Chapter 2

# Relating Attractors and Singular Steady States in the Logical Analysis of Bioregulatory Networks

Heike Siebert and Alexander Bockmayr.

In Proceedings of the 2nd International Conference on *Algebraic Biology*, AB 2007, Castle of Hagenberg, Austria, volume 4545 of *LNCS*, pages 36-50. Springer, 2007.

**Remarks.** At the beginning of my postdoctoral work, Alexander Bockmayr introduced me to the logic-based modeling formalism of René Thomas, and, in particular, to the idea of explicitly including threshold values of interactions in the mathematical representation. In this paper, I adapted the Thomas formalism for Boolean models to include threshold values, using so-called singular states. I particularly focussed on the relation of singular steady states to attractors of the network. Alexander Bockmayr provided helpful comments for the final version of the paper.

# Relating Attractors and Singular Steady States in the Logical Analysis of Bioregulatory Networks

Heike Siebert and Alexander Bockmayr

*Abstract.* In 1973 R. Thomas introduced a logical approach to modeling and analysis of bioregulatory networks. Given a set of Boolean functions describing the regulatory interactions, a state transition graph is constructed that captures the dynamics of the system. In the late eighties, Snoussi and Thomas extended the original framework by including singular values corresponding to interaction thresholds. They showed that these are needed for a refined understanding of the network dynamics. In this paper, we study systematically singular steady states, which are characteristic of feedback circuits in the interaction graph, and relate them to the type, number and cardinality of attractors in the state transition graph. In particular, we derive sufficient conditions for regulatory networks to exhibit multistationarity or oscillatory behavior, thus giving a partial converse to the well-known Thomas conjectures.

## 1 Introduction

Suggested more than 30 years ago, the logical approach to modeling bioregulatory networks has become increasingly popular in the recent past. In the Boolean setting, components of the networks correspond to variables, which can take the values 0 and 1. Interactions between the components are described by logical equations capturing the evolution of the system. R. Thomas contributed a number of papers on the logical analysis of biological networks, starting with [10]. The distinctive feature of his method is the way he derives a representation of the dynamics from the given Boolean functions. Rather than executing all indicated changes in the components at the same time, an asynchronous updating rule is employed to obtain a non-deterministic state transition graph. It has been shown that this approach captures essential qualitative features of the dynamical behavior of complex biological networks, see [11] and [12] for an overview.

In the following years the framework was extended to allow not only for Boolean but multivalued variables that describe different activity levels of the regulatory components in the network. Each interaction in the network was associated with a unique threshold value, which determines when the interaction becomes effective. Snoussi and Thomas realized that a closer inspection of the impact of the threshold values, which they called singular values, would further improve the understanding of the system's dynamics. In [8] they introduced the notion of singular steady states and linked them to feedback circuits in the interaction graph describing the structure of the network. The importance of feedback circuits for the analysis of the dynamical behavior has long been recognized. Thomas conjectured in 1981 that the existence of a positive (resp. negative) circuit, in the interaction graph is a necessary condition for the existence of

two distinct attractors (resp. a cyclic attractor) in the state transition graph. The conjectures have been proven in different settings (see e.g. [9], [4] and [5]). In [2] it is shown, that isolated elementary regulatory circuits result in fundamentally different dynamics depending on their sign. A positive circuit can be linked to the occurrence of two stable states, while a negative circuit causes an attractor comprising dynamical cycles. However, the situation becomes more difficult to grasp as soon as the circuits are embedded in larger and more complex networks.

When trying to incorporate Snoussi's and Thomas' idea of singular states in a Boolean framework, we are faced with several difficulties. On this level of abstraction, every interaction is associated with the same threshold value, a symbolic value between 0 and 1. Thus when crossing the threshold we do not have the advantage of knowing that one and only one interaction becomes effective. As a result we cannot link singular states to circuits in the interaction graph in a non-ambiguous way, while still preserving some essential features known from the multi-valued setting. Despite those complications and the high level of abstraction, this paper shows that the introduction of singular states in the Boolean case is a useful tool for refining our understanding of the relation between structure and dynamics of bioregulatory networks.

The organization of the paper is as follows. In Section 2 we give a short overview of the Boolean description of biological networks and introduce the notion of an attractor of a state transition graph. In Section 3 we extend the framework by establishing the concept of singular states. We give different characterizations of singular steady states using the notion of circuit characteristic states and regular adjacent states. In the main section of this paper, we prove several statements that allow us to derive information on the attractors of the state transition graph from the existence of singular steady states. Conversely, we can deduce the existence of a singular steady state if we have specific knowledge about the attractors of the state transition graph. We conclude by outlining ideas for future work.

## 2 Structure and Dynamics of Regulatory Networks

In the following we introduce the Boolean formalism of R. Thomas for modeling regulatory networks (see for example [11]). We mainly use the notation introduced in [1] and [6]. Throughout the text $\mathcal{B}$ will denote the set $\{0, 1\}$.

**Definition 2.1.** *An* interaction graph *(or bioregulatory graph) $\mathcal{I}$ is a labeled directed graph with vertex set $V := \{\alpha_1, \ldots, \alpha_n\}$, $n \in \mathbb{N}$, and edge set $E$. Each edge $\alpha_j \to \alpha_i$ is labeled with a sign $\varepsilon_{ij} \in \{+, -\}$.*

The only information on a regulatory component we incorporate in the model for now is whether or not it is active. A vertex $\alpha_i$ can be seen as a variable that adopts values in $\mathcal{B}$, where the value 1 indicates that $\alpha_i$ is active. To simplify notation, we identify each vertex $\alpha_i$ with its index $i$.

An edge $\alpha_j \to \alpha_i$ signifies that $\alpha_j$ influences $\alpha_i$ in a positive or negative way depending on the sign $\varepsilon_{ij}$. For each $\alpha_i$ we denote by $Pred(\alpha_i)$ the set of *predecessors* of $\alpha_i$, i. e., the set of vertices $\alpha_j$ such that $\alpha_j \to \alpha_i$ is an edge in $E$.

We will be mainly interested in the following structures of the interaction graph. A tuple $(\alpha_{i_1}, \ldots, \alpha_{i_k})$ of distinct vertices of $\mathcal{I}$ is called a *circuit* if $\mathcal{I}$ contains an edge from $\alpha_{i_j}$ to $\alpha_{i_{j+1}}$ for all $j \in \{1, \ldots, k-1\}$ as well as an edge from $\alpha_{i_k}$ to $\alpha_{i_1}$. The *sign* of a circuit is the product of the sign of its edges.

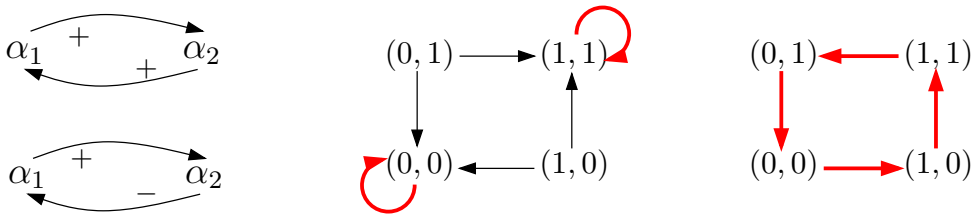



Figure 1: Two interaction graphs consisting of a positive resp. a negative circuit. In both cases we choose $K_{1,\{2\}} = K_{2,\{1\}} = 1$ and $K_{1,\emptyset} = K_{2,\emptyset} = 0$. The state transition graph corresponding to the positive circuit is in the middle, the one corresponding to the negative circuit is on the right. Attractors are indicated by colored, fat lines.

Definition 2.1 captures structural aspects of the network. Now we consider the corresponding dynamical behavior.

**Definition 2.2.** *Let $\mathcal{I}$ be an interaction graph comprising $n$ vertices. A* state *of the system described by $\mathcal{I}$ is a tuple $s \in \mathcal{B}^n$. The set of (regular)* resources $R_i(s) = R_i^{\mathcal{I}}(s)$ *of $\alpha_i$ in state $s$ is the set*

$$\{\alpha_j \in Pred(\alpha_i) \mid (\varepsilon_{ij} = + \wedge s_j = 1) \vee (\varepsilon_{ij} = - \wedge s_j = 0)\}.$$

*Given a set*

$$K(\mathcal{I}) := \{K_{i,\omega} \mid i \in \{1, \ldots, n\}, \omega \subseteq Pred(\alpha_i)\}$$

*of* (logical) parameters, *which adopt values in $\mathcal{B}$, we define the Boolean function $f = f^{K(\mathcal{I})} : \mathcal{B}^n \to \mathcal{B}^n$, $s \mapsto (K_{1,R_1(s)}, \ldots, K_{n,R_n(s)})$. The pair $N := (\mathcal{I}, f)$ is called* bioregulatory network.

The set of resources $R_i(s)$ provides information about the presence of activators and the absence of inhibitors for some regulatory component $\alpha_i$ in state $s$. It contains all genes that contribute to an activation of $\alpha_i$ in state $s$. Note that the absence of an inhibitor is interpreted as an activating influence on the target gene. The value of the parameter $K_{i,R_i(s)}$ indicates how the level of activity $\alpha_i$ will evolve. It will increase ( resp. decrease) if the parameter value is greater (resp. smaller) than $s_i$. The activity level stays the same if both values are equal. Thus, the function $f$ maps a state $s$ to the state the system tends to evolve to. Snoussi and Thomas posed the following condition on the parameter values of the systems they considered:

$$\omega \subseteq \omega' \Rightarrow K_{i,\omega} \leq K_{i,\omega'} \tag{1}$$

for all $i \in \{1, \ldots, n\}$. The condition signifies that an effective activator or a non-effective inhibitor cannot induce the decrease of the activity level of $\alpha_i$. In the following we always assume that this condition is valid.

The choice of parameters completes the definition of the model given by the graph $\mathcal{I}$. Depending on their values, edges in the graph may or may not be *functional* in the following sense. Clearly, if there is an edge $\alpha_j \to \alpha_i$ and $K_{i,M} = K_{i,M\setminus\{\alpha_j\}}$ for all $M \subseteq Pred(\alpha_i)$, then the edge $\alpha_j \to \alpha_i$ has no influence on the dynamics of the system. Eliminating this edge from the interaction graph does not change the function $f$. Thus we may assume for every $N = (\mathcal{I}, f)$ that whenever there is an edge $\alpha_j \to \alpha_i$ in $\mathcal{I}$, there exists a set $M \subseteq Pred(\alpha_i)$ such that $K_{i,M} \neq K_{i,M\setminus\{\alpha_j\}}$.

To derive the dynamics of the system from the function $f$ we take the following consideration into account. In a biological system, the time delays corresponding to changes in the activity level of distinct components will most likely differ. Thus we may assume that in each state transition at most one component is modified. This procedure is called *asynchronous update* in Thomas' framework. We obtain the following definition.

**Definition 2.3.** *The* state transition graph $\mathcal{S}_N$ *describing the dynamics of the network* $N$ *is a directed graph with vertex set* $\mathcal{B}^n$*. There is an edge* $s \to s'$ *if and only if* $s' = f(s) = s$ *or* $s'_i = f_i(s)$ *for some* $i \in \{1, \dots, n\}$ *satisfying* $s_i \neq f_i(s)$ *and* $s'_j = s_j$ *for all* $j \neq i$*.*

In the following we introduce some basic structures in this graph that are of biological interest. In addition we use standard terminology from graph theory, such as paths and cycles.

**Definition 2.4.** *Let* $\mathcal{S}_N$ *be a state transition graph. An infinite path* $(s_0, s_1, \dots)$ *in* $\mathcal{S}_N$ *is called* trajectory*. A nonempty set of states* $D$ *is called* trap set *if every trajectory starting in* $D$ *never leaves* $D$*. A trap set* $A$ *is called* attractor *if for any* $s^1, s^2 \in A$ *there is a path from* $s^1$ *to* $s^2$ *in* $\mathcal{S}_N$*. A state* $s^0$ *is called* steady state*, if* $s^0$ *is a fixed point of* $f$*, that is, if there is an edge from* $s^0$ *to itself. A cycle* $C := (s^1, \dots, s^r, s^1)$*,* $r \geq 2$*, is called a* trap cycle *if every* $s^j$*,* $j \in \{1, \dots, r\}$*, has only one outgoing edge in* $\mathcal{S}_N$*, i. e., the trajectory starting in* $s^1$ *is unique.*

Thus, the attractors of $\mathcal{S}_N$ correspond to the terminal strongly connected components of the graph. It is easy to see that steady states and trap cycles are attractors. In Figure 1 we show two simple interaction graphs. The positive circuit generates a state transition graph with two steady states. The graph derived from the negative circuit consists of a trap cycle, that is, we find an attractor of cardinality greater than one. This corresponds to the typical behavior assigned to positive (resp. negative) circuits mentioned in the introduction.

Attractors represent regions of predictability and stability in the behavior of the system. It is not surprising that an attractor often has a biological interpretation. A fixed point in a gene regulatory network associated with cell differentiation, for example, may represent the stable state reached at the end of a developmental process. Attractors of cardinality greater than one imply cyclic behavior, and thus can often be identified with homeostasis of sustained oscillatory activity, as can be found in the cell cycle or circadian rhythm.

The following proposition is an easy observation concerning attractors.

**Proposition 2.5.** *Every state transition graph* $\mathcal{S}_N$ *contains at least one attractor.*

*Proof.* For $s \in \mathcal{B}^n$ we denote by $D(s)$ the set of states reachable from $s$ by a path in $\mathcal{S}_N$. Then $D(s)$ is a trap set for every $s \in \mathcal{B}^n$. Fix $s \in \mathcal{B}^n$ and choose $A \subseteq D(s)$ a minimal trap set, i. e., every proper subset of $A$ is not a trap set. Let $x, y \in A$. Then $D(x) \subseteq A$, since $A$ is a trap set. Since $A$ is minimal, we have $A = D(x)$. Consequently, there is a path from $x$ to $y$. Thus, $A$ is an attractor. □

Note that the above proof shows that for every state in the state transition graph there is a trajectory leading to an attractor.

The number of states in the state transition graph grows exponentially with the number of regulatory components in $N$. Thus our aim is to infer from restrictions of $f$ to sets of vertices obtained by considering certain subgraphs of $\mathcal{I}$ as much information on the structure of $\mathcal{S}_N$ as possible.

# 3 Singular States

In the following, we incorporate threshold values of interactions into the formalism to get a more complete understanding of the dynamics of the system. We mainly use the framework introduced in [6].

**Definition 3.1.** *Set $\mathcal{B}_\theta := \{0, \theta, 1\}$, where $\theta$ is a symbolic representation of the threshold value and satisfies the order $0 < \theta < 1$. We allow each regulatory component $\alpha_i$ to take values in $\mathcal{B}_\theta$. The values 0 and 1 are called* regular values *and $\theta$ is called* singular value. *The elements of $\mathcal{B}_\theta^n$ are called* states. *If a state comprises only regular components it is called* regular state. *Otherwise it is called* singular state. *For every state $s$ we define $J(s) := \{i \in \{1, \dots, n\} \mid s_i = \theta\}$.*

To describe the dynamics of the system we have to extend the definition of resources.

**Definition 3.2.** *Let $s \in \mathcal{B}_\theta^n$. In addition to the set $R_i(s)$ of regular resources introduced in Definition 2.2, we define the set $R_i^\theta(s)$ of* singular resources *of $\alpha_i$ in $s$ as the set*

$$R_i^\theta(s) := \{\alpha_j \in Pred(\alpha_i) \mid s_j = \theta\}.$$

The definition of a set of logical parameters $K(\mathcal{I})$ remains the same as in Definition 2.2. In particular, the logical parameters can only adopt regular values.

We call $|a, b|$ a *qualitative value* if $a, b \in \mathcal{B}$ and $a \leq b$. The qualitative value $|0, 0|$ is identified with the regular value 0, $|1, 1|$ with the regular value 1, and $|0, 1|$ with the singular value $\theta$. The relations $<$, $>$, and $=$ are used with respect to this identification.

**Definition 3.3.** *Let $K(\mathcal{I})$ be a set of parameters. We define*

$$f^\theta = f^{K(\mathcal{I}),\theta} : \mathcal{B}_\theta^n \to \mathcal{B}_\theta^n \quad \text{by} \quad f_i^\theta(s) = |K_{i,R_i(s)}, K_{i,R_i(s)\cup R_i^\theta(s)}|$$

*for all $i \in \{1, \dots, n\}$.*

The map $f^\theta$ is well defined since condition (1) ensures that $K_{i,R_i(s)} \leq K_{i,R_i(s)\cup R_i^\theta(s)}$ for all $i \in \{1, \dots, n\}$. Note that whenever $s$ is a regular state, then $f^\theta(s)$ is regular, too, since any set of singular resources in a regular state is empty. We have $f^\theta(s) = f(s)$ for all $s \in \mathcal{B}^n$. Thus the state transition graph corresponding to $N = (\mathcal{I}, f)$ is consistent with $f^\theta$. Extending the definition in the previous section, we call $s \in \mathcal{B}_\theta^n$ a *steady state* if $f^\theta(s) = s$. The notion of functionality of an edge remains the same as in Section 2. We consider only those edges that effectively influence the dynamical evolution of the system.

We may relate a singular state $s$ to structures in the interaction graph $\mathcal{I}$ by considering the subgraphs of $\mathcal{I}$ induced by the vertices $\alpha_j$ with singular values, that is $j \in J(s)$. The following definition proves useful and was first introduced by E. H. Snoussi in [8], albeit in a different framework. The remainder of this section adapts ideas presented in [8].

**Definition 3.4.** *Let $C = (\alpha_{i_1}, \dots, \alpha_{i_r})$ be a circuit in $\mathcal{I}$. A state $s \in \mathcal{B}_\theta^n$ is called* characteristic state *of $C$ if $s_{i_l} = \theta$ for all $l \in \{1, \dots, r\}$.*

A characteristic state of a circuit is not unique unless all the regulatory components of the network are contained in the circuit. In this case the state $(\theta, \dots, \theta)$ is the unique characteristic state. Obviously, the state $(\theta, \dots, \theta)$ is characteristic of each circuit in $\mathcal{I}$.

Another simple observation is the following. Whenever $R_j^\theta(s) \neq \emptyset$ holds for all singular components $j \in J(s)$, the state $s$ is characteristic of some circuit in $\mathcal{I}$. This is due to the fact that every resource of some regulatory component $\alpha_i$ is a predecessor of $\alpha_i$ and that there are only finitely many components in the network. With that in mind we can easily prove the next statement.

**Theorem 3.5.** *Every singular steady state is characteristic of some circuit in $\mathcal{I}$.*

*Proof.* Let $s$ be a singular state that is not characteristic of any circuit in $\mathcal{I}$. Then there is $i \in \{1,\dots,n\}$ such that $s_i = \theta$ and $R_i^\theta(s) = \emptyset$. It follows that $f_i^\theta(s) = |K_{i,R_i(s)}, K_{i,R_i(s)}| = K_{i,R_i(s)} \neq \theta = s_i$, since the parameters take only regular values. Thus $s$ is not a steady state. □

If the network consists of a single circuit, then the corresponding characteristic state is always steady under our standard assumption that every edge in the graph is functional. As mentioned before, such a circuit displays a characteristic behavior depending on its sign. In general, the existence of a steady characteristic state of a circuit does not always result in the corresponding dynamical behavior, as will be illustrated in the next section.

It is possible to give a characterization of the singular steady states using only regular states and the function $f$.

**Definition 3.6.** *Let $s \in \mathcal{B}_\theta^n$ and $k \in \{1,\dots,n\}$. Let $s^{k,+}$ and $s^{k,-}$ be regular states that satisfy $s_i^{k,+} := s_i^{k,-} := s_i$ for all $i \notin J(s)$ and*

$$s_i^{k,+} := \begin{cases} 1 & , \quad \varepsilon_{ki} = + \\ 0 & , \quad \varepsilon_{ki} = - \end{cases} \quad \text{and} \quad s_i^{k,-} := \begin{cases} 1 & , \quad \varepsilon_{ki} = - \\ 0 & , \quad \varepsilon_{ki} = + \end{cases} \tag{2}$$

*for all $i \in J(s)$ satisfying $\alpha_i \in R_k^\theta(s)$. Then $s^{k,+}$ and $s^{k,-}$ are called a* maximal *resp.* minimal adjacent state *of $s$ with respect to $k$.*

There are generally many states $s^{k,+}$, $s^{k,-}$ that satisfy the above conditions. If the sets $R_k^\theta(s)$, $k \in \{1,\dots,n\}$, are disjoint, then we can define states $s^+$ and $s^-$ which are maximal resp. minimal adjacent states of $s$ with respect to every $k \in \{1,\dots,n\}$. If, in addition, the union of all sets $R_k^\theta(s)$ is equal to the set $\{\alpha_j\,;\, j \in J\}$, then $s^+$ and $s^-$ are unique and are called the maximal resp. minimal adjacent state of $s$.

**Theorem 3.7.** *A state $s \in \mathcal{B}_\theta^n$ is steady iff for all $k \in \{1,\dots,n\}$ there is some choice of $s^{k,+}$, $s^{k,-}$ such that $f_k(s^{k,+}) = s_k^{k,+} = s_k^{k,-} = f_k(s^{k,-})$, if $k \notin J(s)$, and $f_k(s^{k,-}) < \theta < f_k(s^{k,+})$, if $k \in J(s)$.*

*Proof.* We show that $R_k(s^{k,+}) = R_k(s) \cup R_k^\theta(s)$ and $R_k(s^{k,-}) = R_k(s)$ for all $k \in \{1,\dots,n\}$. First, let $\alpha_i \in R_k(s^{k,+})$. Then $\alpha_i$ is a predecessor of $\alpha_k$. If $i \notin J := J(s)$, then $s_i = s_i^{k,+}$, and thus $\alpha_i \in R_k(s)$. If $i \in J$, we have $s_i = \theta$, and thus $\alpha_i \in R_k^\theta(s)$. Now, let $\alpha_i \in R_k(s) \cup R_k^\theta(s)$. Again $\alpha_i \in Pred(\alpha_k)$. If $\alpha_i \in R_k(s)$, then $i \notin J$. It follows that $s_i = s_i^{k,+}$, and thus $\alpha_i \in R_k(s^{k,+})$. If $\alpha_i \in R_k^\theta(s)$, then $\alpha_i \in R_k(s^{k,+})$ according to (2). Analogous reasoning provides the second statement.

Now, suppose that the last condition of the theorem is true. Then

$$f_k^\theta(s) = |K_{k,R_k(s)}, K_{k,R_k(s)\cup R_k^\theta(s)}| = |K_{k,R_k(s^{k,-})}, K_{k,R_k(s^{k,+})}| = |f_k(s^{k,-}), f_k(s^{k,+})|$$

for all $k \in \{1, \ldots, n\}$. According to the assumption we have $|f_k(s^{k,-}), f_k(s^{k,+})| = s_k^{k,+} = s_k$ for $k \notin J$, and $|f_k(s^{k,-}), f_k(s^{k,+})| = |0,1| = s_k$ for all $k \in J$. Thus $s$ is a steady state. Similar reasoning can be used to show the inverse statement. □

The theorem and the definition of $s^{k,+}$ and $s^{k,-}$ imply that whenever every regulatory component in the network can be influenced in its behavior by some other regulatory components, the state containing only singular entries is a steady state. In other words, if for every $\alpha_k$ we have $K_{\alpha_k,\emptyset} = 0$ and $K_{\alpha_k,Pred(\alpha_k)} = 1$, then the state $(\theta, \ldots, \theta)$ is a steady state.

## 4 Relating Singular Steady States and Attractors

We have seen that singular steady states can be characterized by regular states and that they are closely related to circuits in the interaction graph. In the following we show what kind of information on the state transition graph can be inferred from the existence of a singular steady state. First, we need some additional notations.

Let $s \in \mathcal{B}_\theta^n$ be a singular state. Recall that $J(s)$ is the set of indices corresponding to the singular values of $s$ and that we identify each vertex $\alpha_i$ with its index $i$. With $\mathcal{I}^\theta(s)$ we denote the graph with vertex set $V^\theta(s) := J(s)$ and edge set $E^\theta(s)$ consisting of those $\{\alpha_i, \alpha_j\}$ with $i, j \in J(s)$ such that $\alpha_i \to \alpha_j$ or $\alpha_j \to \alpha_i$ is an edge in $\mathcal{I}$. The graph $\mathcal{I}^\theta(s)$ is undirected. It represents the existence of a dependency between singular components, without specifying the type of interaction. A (connected) *component* of $\mathcal{I}^\theta(s)$ is a maximal connected subgraph of $\mathcal{I}^\theta(s)$. By abuse of notation we denote the vertex set of a component $Z$ of $\mathcal{I}^\theta(s)$ also with $Z$. Vertices of different components of $\mathcal{I}^\theta(s)$ represent regulatory components in $\mathcal{I}$ that do not influence each other directly. Figure 2 illustrates the concept on a small example. Let $C$ be a circuit composed of vertices in $J(s)$. Then there is a component of $\mathcal{I}^\theta(s)$ which contains the vertices of $C$. We denote this component by $J_C(s)$.

The next lemma shows that for a singular steady state $s$ value changes in a component of $\mathcal{I}^\theta(s)$ do not influence the image $f^\theta(s)$ outside that component. It will play an important role in all the following considerations.

**Lemma 4.1.** *Let $s$ be a singular steady state, and let $Z_1, \ldots, Z_m$ be the components of $\mathcal{I}^\theta(s)$. Consider a union $Z$ of arbitrary components $Z_j$. Let $\tilde{s} \in \mathcal{B}_\theta^n$ such that $\tilde{s}_i = s_i$ for all $i \notin Z$. Then $f_i^\theta(\tilde{s}) = f_i^\theta(s) = s_i = \tilde{s}_i$ for all $i \notin Z$.*

*Proof.* For $i \in J(s) \setminus Z$ we know that $R_i(s) = R_i(\tilde{s})$ and $R_i^\theta(s) = R_i^\theta(\tilde{s})$, since no element of $Z$ is a predecessor of $\alpha_i$. Thus $f_i^\theta(\tilde{s}) = f_i^\theta(s) = s_i$ for all $i \in J(s) \setminus Z$. For $i \notin J(s)$ we have $R_i(s) \subseteq R_i(\tilde{s})$, since a singular resource of $\alpha_i$ may have turned into a regular resource. In addition, $R_i(\tilde{s}) \cup R_i^\theta(\tilde{s}) \subseteq R_i(s) \cup R_i^\theta(s)$, since a singular resource of $\alpha_i$ might have been eliminated by turning its value to a regular value not contributing to activation. In summary we obtain $R_i(s) \subseteq R_i(\tilde{s}) \subseteq R_i(\tilde{s}) \cup R_i^\theta(\tilde{s}) \subseteq R_i(s) \cup R_i^\theta(s)$ and with condition (1) we derive

$$K_{i,R_i(s)} \leq K_{i,R_i(\tilde{s})} \leq K_{i,R_i(\tilde{s}) \cup R_i^\theta(\tilde{s})} \leq K_{R_i(s) \cup R_i^\theta(s)}.$$

Moreover, $K_{i,R_i(s)} = K_{i,R_i(s) \cup R_i^\theta(s)}$, since $f_i^\theta(s) = s_i$. Thus the above inequality becomes an equality and $f_i^\theta(\tilde{s}) = K_{i,R_i(s)} = s_i = \tilde{s}_i$ for all $i \notin J(s)$. □

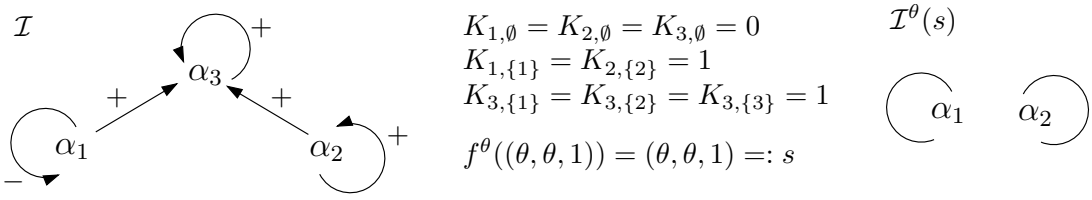



Figure 2: An interaction graph $\mathcal{I}$ and a specification of the parameters. Missing parameter values follow from condition (1). The graph $\mathcal{I}^\theta(s)$ for $s := (\theta, \theta, 1)$ has two components.

The above lemma allows us to focus on the possible dynamical behavior in the isolated parts of the biological network corresponding to the components $Z_1, \ldots, Z_m$ and leads us to the following theorem.

**Theorem 4.2.** *For every singular steady state $s$ there is an attractor $A$ in $\mathcal{S}_N$ such that $u_i = s_i$ holds for all $u \in A$ and $i \notin J(s)$.*

*Proof.* Set $P := \{x \in \mathcal{B}^n \mid \forall i \notin J(s) : x_i = s_i\}$. Then $f_i(x) = x_i = s_i$ for all $i \notin J(s)$ according to Lemma 4.1, i. e., $f(x) \in P$. Thus all successors of $x$ in $\mathcal{S}_N$ are also in $P$. It follows that $P$ is a trap set. Like in the proof of Prop. 2.5 we deduce that $P$ contains an attractor $A$, and $u_i = s_i$ for all $u \in A$ and $i \notin J(s)$. □

It is not difficult to see that we can derive such an attractor $A$ from attractors $A_1, \ldots, A_k$ arising in the system's dynamical behavior restricted to the components $Z_1, \ldots, Z_k$ of $\mathcal{I}^\theta(s)$. To illustrate this we examine the example given in Figure 2. The state $(\theta, \theta, 1)$ is steady, the components of $\mathcal{I}^\theta(s)$ are $Z_1 = \{\alpha_1\}$ and $Z_2 = \{\alpha_2\}$. We consider the dynamics restricted to $Z_1$ given by the projection $f^{(Z_1)} : \mathcal{B} \to \mathcal{B}, x \mapsto f_1^\theta(x, \theta, 1)$. It generates a state transition graph that consists of a cycle comprising the states 0 and 1. Thus it has a single attractor $A_1 = \{0, 1\}$. The state transition graph corresponding to the analogously defined function $f^{(Z_2)}$ consists of the two attractors $A_2^1 = \{0\}$ and $A_2^2 = \{1\}$. According to Lemma 4.1, the value of the third component of $s$ will remain fixed, regardless of the values of the first two components. Thus we can derive two attractors in $\mathcal{S}_N$, namely $A^1 = A_1 \times A_2^1 \times \{s_3\} = \{(0, 0, 1), (1, 0, 1)\}$ and $A^2 = A_1 \times A_2^2 \times \{s_3\} = \{(0, 1, 1), (1, 1, 1)\}$.

We have seen above that we can link a singular steady state to a regular attractor. However, different singular steady states $s^1$ and $s^2$ may give rise to the same regular attractor. The above proof shows that this possibility is precluded if $s^1$ and $s^2$ differ in a component $i \notin J(s^1) \cup J(s^2)$.

A more precise analysis of the correspondence of attractors and singular steady states is possible if we take into account structural information on the underlying interaction graph $\mathcal{I}$. In the preceding section we have seen that every singular steady state $s$ is characteristic of some circuit $C$ of the interaction graph $\mathcal{I}$. If we know in addition that $s$ is not characteristic of any other circuit in $\mathcal{I}$ with vertices in the connected component $J_C(s)$ of $\mathcal{I}^\theta(s)$, we can derive information on the singular valued predecessors of vertices belonging to $C$. This is shown in the next lemma.

**Lemma 4.3.** *Let $C = (\alpha_{i_1}, \ldots, \alpha_{i_m})$ be a circuit in $\mathcal{I}$ and let $s$ be a steady characteristic state of $C$. Assume that $C$ is the only circuit in $\mathcal{I}$ with all its vertices contained in $J_C(s)$. Then $R_{i_j}^\theta(s) = \{\alpha_{i_{j-1}}\}$ for all $j \in \{1, \ldots, m\}$ with indices taken modulo $m$.*

*Proof.* Set $J := J(s)$ and $J_C := J_C(s)$. Clearly, $\alpha_{i_{j-1}} \in R^\theta_{i_j}(s)$ for all $j \in \{1,\dots,m\}$. Assume that there is $k \in \{1,\dots,m\}$ such that there exists $l \in J$ satisfying $\alpha_l \neq \alpha_{i_{k-1}}$ and $\alpha_l \in R^\theta_{i_k}(s)$. Then $\alpha_l \in Pred(\alpha_{i_k})$ and thus $l \in J_C$. If $l = i_j$ for some $j \neq k-1$, then $(\alpha_{i_j}, \alpha_{i_k}, \dots, \alpha_{i_{j-1}})$ is a circuit other than $C$ in $J_C$. This contradicts the hypothesis. Thus $\alpha_l$ is not a vertex of $C$.

Since $s$ is a steady state, we know that $R^\theta_j(s) \neq \emptyset$ for all $j \in J$. Furthermore, $R^\theta_j(s) \subseteq J_C$ for all $j \in J_C$. Thus for every $j \in J_C$ we find $i \in J_C$, such that $\alpha_i \to \alpha_j$ is an edge in $\mathcal{I}$. Since there are only finitely many vertices in $J_C$, there is a circuit in $\{\alpha_j \in J_C\,;\, \exists \text{ path from } \alpha_j \text{ to } \alpha_l \text{ in } \mathcal{I}\}$ that differs from $C$. Again, this leads to a contradiction. □

Note that there may be vertices in $J_C(s)$ that have more than one singular resource. Lemma 4.3 allows us to represent $J_C(s)$ by a chain of nested sets.

**Lemma 4.4.** *Under the hypotheses of Lemma 4.3, there exist sets $M_1,\dots,M_l \subseteq J_C(s)$ such that $M_1 = \{i_1,\dots,i_m\}$, $M_l = J_C(s)$, $M_i \subsetneq M_{i+1}$ and $R^\theta_j(s) \subseteq M_i$ for all $j \in M_{i+1}$ and $i \in \{1,\dots,l-1\}$.*

*Proof.* Set $M_1 := \{i_1,\dots,i_m\}$. If $J_C(s) \setminus M_1 \neq \emptyset$, then there exists at least one element $j \in J_C(s)\setminus M_1$ such that $R^\theta_j(s) \subseteq M_1$. Otherwise for every $j \in J_C(s)\setminus M_1$ there is $k_j \in J_C(s)\setminus M_1$ such that $\alpha_{k_j}$ is a predecessor of $\alpha_j$ in $\mathcal{I}$. That would imply the existence of a circuit other than $C$ in $J_C(s)$, since $J_C(s) \setminus M_1$ is finite. Thus by defining $M_2 := \{j \in J_C(s)\,;\, R^\theta_j(s) \subseteq M_1\}$ we obtain a set strictly containing $M_1$. Since $J_C(s)$ is finite, we can repeat the procedure until we get $M_l := \{j \in J_C(s)\,;\, R^\theta_j(s) \subseteq M_{l-1}\} = J_C(s)$. □

In the following we make use of the information on the sign of the circuit $C$.

**Theorem 4.5.** *Let $C$ be a positive circuit in $\mathcal{I}$ and let $s$ be a steady characteristic state of $C$. Assume that $C$ is the only circuit in $\mathcal{I}$ with all its vertices contained in $J_C(s)$. Then $f^\theta$ has at least three fixed points.*

*Proof.* Set $J := J(s)$ and $J_C := J_C(s)$. Without loss of generality we may assume that $C = (\alpha_1,\dots,\alpha_r)$ for some $r \in \{1,\dots,n\}$. We determine states $s^0, s^1 \in \mathcal{B}^n_\theta$ by an iterative process such that $s$, $s^0$ and $s^1$ are fixed points of $f^\theta$. Initially, we set $s^0_i := s^1_i := s_i$ for all $i \notin J_C$ and choose the other components of $s^0$ and $s^1$ arbitrary.

From Lemma 4.1 it follows that $f^\theta_i(s^0) = s^0_i$ and $f^\theta_i(s^1) = s^1_i$ for all $i \notin J_C$. Next, we define the values $s^0_i$ and $s^1_i$ for $i \in \{1,\dots,r\}$. We set $s^0_1 := 0$, $s^1_1 := 1$, and for $l \in \{0,1\}$

$$s^l_{i+1} := \begin{cases} 0 & , \quad (s^l_i = 0 \wedge \varepsilon_{i+1,i} = +) \vee (s^l_i = 1 \wedge \varepsilon_{i+1,i} = -) \\ 1 & , \quad (s^l_i = 1 \wedge \varepsilon_{i+1,i} = +) \vee (s^l_i = 0 \wedge \varepsilon_{i+1,i} = -) \end{cases}$$

for all $i \in \{1,\dots,r-1\}$. This definition amounts to setting $s^l_{i+1} = 1$ iff the value of $s^l_i$ characterizes $\alpha_i$ as regular resource of $\alpha_{i+1}$. As is easy to see we also have

$$s^l_{i+1} = \begin{cases} s^l_i & , \quad \varepsilon_{i+1,i} = + \\ 1 - s^l_i & , \quad \varepsilon_{i+1,i} = - \end{cases} .$$

It follows for all $i \in \{1,\dots,r-1\}$ that $s^l_{i+1} = s^l_1$ if $\prod_{j=1}^i \varepsilon_{j+1,j}$ is positive, and $s^l_{i+1} = s^l_1$ if $\prod_{j=1}^i \varepsilon_{j+1,j}$ is negative. Since $C$ is a positive circuit, the value of $s^l_1$ is consistent with the value we obtain by using the above definition for $i = r$, that is we do not contradict the definition of $s^l$ if we use the above iterative formula modulo $r$. Note that $s^0_1$, $s^1_1$ and $s_1$ are distinct.

According to Lemma 4.3 we have $R_i^\theta(s) = \{\alpha_{i-1}\}$ for all $i \in \{1,\dots,r\}$, indices again taken modulo $r$. Thus $R_i^\theta(s^l) = \emptyset$ for all $i \in \{1,\dots,r\}$. Moreover, we have

$$R_i(s^l) = \begin{cases} R_i(s) & , \ (s^l_{i-1} = 0 \wedge \varepsilon_{i,i-1} = +) \vee (s^l_{i-1} = 1 \wedge \varepsilon_{i,i-1} = -) \\ R_i(s) \cup R_i^\theta(s) & , \ (s^l_{i-1} = 1 \wedge \varepsilon_{i,i-1} = +) \vee (s^l_{i-1} = 0 \wedge \varepsilon_{i,i-1} = -) \end{cases}$$

for all $i \in \{1,\dots,r\}$. Since $f_i^\theta(s) = |K_{i,R_i(s)}, K_{i,R_i(s)\cup R_i^\theta(s)}| = |0,1|$, it follows from the definition of $s_i^l$ and condition (1) that

$$f_i^\theta(s^l) = K_{i,R_i(s^l)} = \begin{cases} K_{i,R_i(s)} = 0 & , \quad s_i^l = 0 \\ K_{i,R_i(s)\cup R_i^\theta(s)} = 1 & , \quad s_i^l = 1 \end{cases} .$$

Thus, we have $f_i^\theta(s^0) = s_i^0$ and $f_i^\theta(s^1) = s_i^1$ for all $i \in \{1,\dots,r\}$, not depending on the values of the components in $J_C \setminus \{1,\dots,r\}$.

Finally, we have to specify $s_i^l$ for all $i \in J_C \setminus \{1,\dots,r\}$ and $l \in \{0,1\}$. According to Lemma 4.4 we find sets $M_1,\dots,M_k \subseteq J_C$ satisfying $M_1 = \{1,\dots,r\}$, $M_k = J_C$, $M_j \subsetneq M_{j+1}$ and $R_i^\theta(s) \subseteq M_j$ for all $i \in M_{j+1}$ and $j \in \{1,\dots,k-1\}$. Thus we can deduce that $\alpha_i$, $i \in M_2$, has no predecessors in $J_C \setminus M_1$, since otherwise they would be in $R_i^\theta(s)$. Furthermore, for every $i \in M_2$ we have $R_i^\theta(s^l) = \emptyset$ since all components corresponding to vertices in $C$ have regular values. Now we set $s_i^l := K_{i,R_i(s^l)}$ for all $i \in M_2$. Note that this parameter depends only on components previously specified, i. e., on the values $s_i^l$ for $i \notin J_C \setminus \{1,\dots,r\}$. Since $\alpha_i$ does not have singular resources in state $s^l$ for all $i \in M_2$, we have $f_i^\theta(s^l) = K_{i,R_i(s^l)} = s_i^l$ for all $i \in M_2$. Because the sets $M_j$ are nested, we can repeat the above procedure for consecutive sets without encountering contradictions. Thus we are able to specify all components $s_i^l$ for $i \in J_C \setminus \{1,\dots,r\}$, such that $f_i^\theta(s^l) = s_i^l$.

We have shown that the resulting states $s^0$ and $s^1$ are fixed points of $f^\theta$. Since $s$, $s^0$, and $s^1$ are distinct, $f^\theta$ has at least three fixed points. □

The proof shows that at least two fixed points of $f^\theta$ differ in a regular component. Applying Theorem 4.2 and the subsequent observations we immediately obtain the following statement.

**Corollary 4.6.** *Under the hypotheses of Theorem 4.5 there are at least two distinct attractors in the corresponding state transition graph.*

The corollary is illustrated in Figure 3 (a) and (c). The singular steady state $(1,\theta,0)$ is characteristic of the positive circuit comprising $\alpha_2$ and of no other circuit. The resulting state transition graph shows two distinct attractors. The importance of the condition concerning the circuit $C$ and the component $J_C(s)$ is demonstrated in Figure 3 (b). The state $(\theta,\theta,\theta)$ is steady and characteristic of the positive circuit comprising $\alpha_2$. Moreover, the state $(\theta,0,\theta)$ is steady and characteristic of the positive circuit comprising $\alpha_1$ and $\alpha_3$. In both cases the states are characteristic of further circuits in the same component, and the state transition graph has only one attractor. Figure 4 shows the importance of $C$ being the only circuit with vertices in $J_C(s)$ for the validity of Theorem 4.5. The interaction graph given in (a) contains a positive circuit with characteristic state $s := (\theta,\theta,\theta,\theta)$. Together with the parameters given in (b) it gives rise to a system that has no regular fixed point. Moreover, from the logical implications in (d) we can easily deduce that $s$ is the only singular steady state.

The network in Figure 4 (b), together with the parameters given in (c), illustrates that the sufficient condition of Theorem 4.5 is not necessary. The given system has two regular fixed

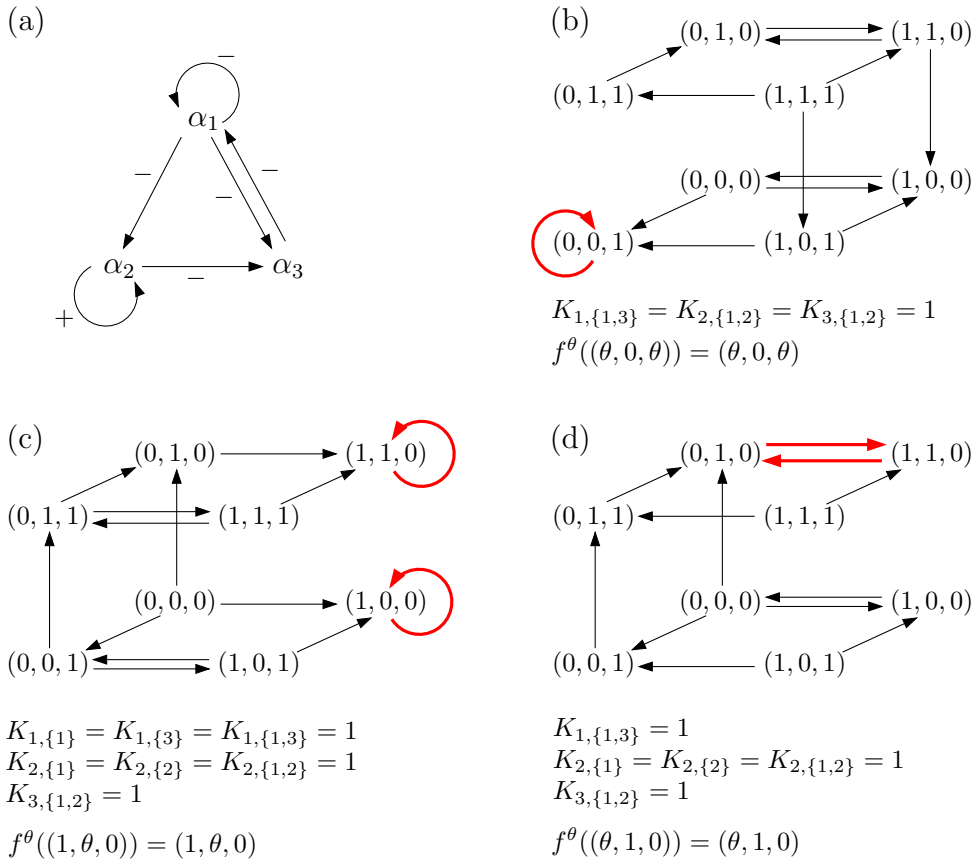



Figure 3: An interaction graph comprising three components is given in (a). Figures (b)-(d) show the state transition graphs corresponding to the chosen parameter values. We only listed the non-zero parameters. Attractors are indicated by colored, fat lines. For each choice of parameters one singular steady state other than $(\theta,\theta,\theta)$ is given.

points, (0,0,0,0) and (1,1,1,1). However, the only steady characteristic state is $s := (\theta,\theta,\theta,\theta)$, as easy to see from the implications in (d). Its components comprise the vertices of all three cycles of the network.

The next theorem clarifies the impact of a negative circuit.

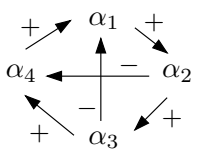
**Theorem 4.7.** *Let $C$ be a negative circuit in $\mathcal{I}$ and let $s$ be a steady characteristic state of $C$. Assume that $C$ is the only circuit in $\mathcal{I}$ with all its vertices contained in $J_C(s)$. Then there exists an attractor with cardinality greater than one.*

*Proof.* Again set $J := J(s)$ and $J_C := J_C(s)$ and assume that $C = (\alpha_1,\dots,\alpha_r)$ for some $r \in \{1,\dots,n\}$. By $P_j$, $j \in \{1,\dots,r\}$, we denote the set of all regular states $x$ satisfying $x_k = s_k$ for all $k \notin J$ and

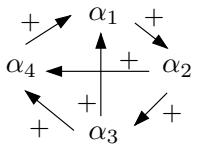
$$x_{i+1} = \begin{cases} x_i & , \quad \varepsilon_{i+1,i} = + \\ 1 - x_i & , \quad \varepsilon_{i+1,i} = - \end{cases} \qquad \text{for all } i \in \{1,\dots,r\} \setminus \{j\},$$

with indices $i$ taken modulo $r$. Choose $j \in \{1,\dots,r\}$ and $x \in P_j$. Lemma 4.1 implies that $f_i(x) = s_i$ for all $i \notin J$. Now set $\tilde{x} = f(x)$ and let $i \in \{1,\dots,r\}$. Again, consider indices modulo

(a)




(b)




(c)

$$\begin{array}{lcl} K_{\alpha_1,\{\alpha_3\}} & = & 1 \\ K_{\alpha_1,\{\alpha_4\}} & = & 1 \\ K_{\alpha_1,\{\alpha_3,\alpha_4\}} & = & 1 \\ K_{\alpha_2,\{\alpha_1\}} & = & 1 \\ K_{\alpha_3,\{\alpha_2\}} & = & 1 \\ K_{\alpha_4,\{\alpha_2,\alpha_3\}} & = & 1 \end{array}$$

(d)

$$\begin{array}{lcl} s_1 = \theta & \Rightarrow & f_2^\theta(s) = \theta \\ s_1 \neq \theta & \Rightarrow & f_2^\theta(s) \neq \theta \\ s_2 = \theta & \Rightarrow & f_3^\theta(s) = \theta,\ f_4^\theta(s) = 0, \theta \\ s_2 \neq \theta & \Rightarrow & f_3^\theta(s) \neq \theta \\ s_3 = \theta & \Rightarrow & f_1^\theta(s) = \theta, 1,\ f_4^\theta(s) = 0, \theta \\ s_4 = \theta & \Rightarrow & f_1^\theta(s) = \theta, 1 \end{array}$$

$$s \text{ singular, } f^\theta(s) = s \quad \Rightarrow \quad s = (\theta, \theta, \theta, \theta)$$

Figure 4: Interaction graphs and parameter values of networks with only one singular steady state. Given are the non-zero logical parameters. For details see the text.

$r$. According to Lemma 4.3 the only singular resource of $\alpha_{i+1}$ in $s$ is $\alpha_i$. Furthermore, we know $f_{i+1}^\theta(s) = s_{i+1} = \theta = |0,1|$. Thus, with reasoning similar to that in the proof of Theorem 4.5, we can deduce that

$$\tilde{x}_{i+1} = K_{i+1,R_{i+1}(x)} = \left\{ \begin{array}{lcl} 0 & , & (x_i = 0 \wedge \varepsilon_{i+1,i} = +) \vee (x_i = 1 \wedge \varepsilon_{i+1,i} = -) \\ 1 & , & (x_i = 1 \wedge \varepsilon_{i+1,i} = +) \vee (x_i = 0 \wedge \varepsilon_{i+1,i} = -) \end{array} \right. ,$$

that is

$$\tilde{x}_{i+1} = \left\{ \begin{array}{lcl} x_i & , & \varepsilon_{i+1,i} = + \\ 1 - x_i & , & \varepsilon_{i+1,i} = - \end{array} \right. .$$

Now, if $i \neq j+1$, we can express $x_i$ in terms of $x_{i-1}$, since $x$ is in $P_j$. Furthermore, we can then express $x_{i-1}$ in terms of $\tilde{x}_i$ according to the observation above, which is valid for all $i \in \{1, \ldots, r\}$. Some easy substitutions yield firstly

$$\tilde{x}_{i+1} = \left\{ \begin{array}{lcl} x_{i-1} & , & (\varepsilon_{i+1,i} = + \wedge \varepsilon_{i,i-1} = +) \vee (\varepsilon_{i+1,i} = - \wedge \varepsilon_{i,i-1} = -) \\ 1 - x_{i-1} & , & (\varepsilon_{i+1,i} = + \wedge \varepsilon_{i,i-1} = -) \vee (\varepsilon_{i+1,i} = - \wedge \varepsilon_{i,i-1} = +) \end{array} \right. ,$$

and secondly that $\tilde{x}_{i+1} = \tilde{x}_i$, if $\varepsilon_{i+1,i} = +$, and $\tilde{x}_{i+1} = 1 - \tilde{x}_i$, if $\varepsilon_{i+1,i} = -$. It follows that $\tilde{x} = f(x)$ is an element of $P_{j+1}$. Furthermore, in case $\varepsilon_{i,i-1} = +$ and $i \neq j+1$, we have $\tilde{x}_i = x_{i-1}$ as seen above and $x_{i-1} = x_i$, since $x \in P_j$. This shows $f_i(x) = \tilde{x}_i = x_i$. The same reasoning leads to $f_i(x) = \tilde{x}_i = x_i$ for $\varepsilon_{i,i-1} = -$ and $i \neq j+1$. It follows that every successor $x'$ of $x$ in the state transition graph is either in $P_j$, in case $x'_{j+1} = x_{j+1}$, or in $P_{j+1}$, in case $x'_{j+1} \neq x_{j+1}$. Since our reasoning is true for indices modulo $r$, we can deduce that the union $P$ of the sets $P_j$, $j \in \{1, \ldots, r\}$, is a trap set and thus contains an attractor $A$ (see the proof of Proposition 2.5).

Finally, we show that each state in $P$, and thus in $A$, has a successor other than itself. For $x \in P_j$ we have

$$x_j = \left\{ \begin{array}{lcl} x_{j+1} & , & \varepsilon_{j+2,j+1} \cdot \ldots \cdot \varepsilon_{j,j-1} = + \\ 1 - x_{j+1} & , & \varepsilon_{j+2,j+1} \cdot \ldots \cdot \varepsilon_{j,j-1} = - \end{array} \right. .$$

Furthermore, we know that $\tilde{x}_j = x_j$ with $\tilde{x} := f(x)$ and $\tilde{x} \in P_{j+1}$. It follows that $\tilde{x}_{j+1} = x_j$, if $\varepsilon_{j+1,j} = +$, and $\tilde{x}_{j+1} = 1 - x_j$, if $\varepsilon_{j+1,j} = -$. Thus we obtain

$$\tilde{x}_{j+1} = \left\{ \begin{array}{lcl} x_{j+1} & , & \prod_{k=1}^r \varepsilon_{k+1,k} = + \\ 1 - x_{j+1} & , & \prod_{k=1}^r \varepsilon_{k+1,k} = - \end{array} \right. ,$$

with indices $k$ taken modulo $r$. Since $C$ is negative, we know $\prod_{k=1}^{r} \varepsilon_{k+1,k} = -$, and thus $f_{j+1}(x) \neq x_{j+1}$. Thus $x$ has a successor other than itself in the state transition graph. It follows that the cardinality of $A$ is greater than one. □

Figure 3 illustrates the theorem. In (d) we give a parameter specification that allows the state $(\theta, 1, 0)$ to be steady. This state is characteristic of the negative circuit comprising $\alpha_1$. The resulting state transition graph contains the attractor $\{(0,1,0),(1,1,0)\}$. As for Theorem 4.5, Figure 3 (b) illustrates the importance of $C$ being the only circuit in $J_C(s)$. Although $(\theta, 0, \theta)$ is characteristic of the negative circuit comprising $\alpha_1$, and $(\theta, \theta, \theta)$ is characteristic of the negative circuit comprising $\alpha_1$, $\alpha_2$ and $\alpha_3$, the only attractor in the state transition graph consists of a single state. Figure 4 (a) and (c) specify a system that illustrates that the sufficient condition in Theorem 4.7 is not necessary. By calculating the corresponding state table we can see that there is no regular steady state of the system. Thus there has to be an attractor with cardinality greater than one. However, from the logical implications given in (d), it follows easily that the only singular steady state is $(\theta, \theta, \theta, \theta)$, which is characteristic for all circuits in the interaction graph given in (a).

The proofs of Theorems 4.5 and 4.7 show that the situation is easy to grasp in case that the only components with singular values are those of the circuit $C$. In the context of Theorem 4.5, we then obtain two regular fixed points, that is two steady states in the state transition graph. Those can be explicitly constructed as shown in the proof of Theorem 4.5. If $C$ is a negative circuit, we find a trap cycle in the state transition graph. It is composed of the states in the set $P$ introduced in the proof of Theorem 4.7.

If we detect the above mentioned structures in the state transition graph, we can conversely derive singular steady states. The proofs of the next two propositions are omitted for lack of space. They can be found in [7].

**Proposition 4.8.** *Let $x, y \in \mathcal{B}^n$ be steady states in the state transition graph $\mathcal{S}_N$. Let $I$ be the set of components $i$ satisfying $x_i \neq y_i$. Then there exists a singular steady state $s$ such that $s_i = \theta$ for all $i \in I$.*

**Proposition 4.9.** *Let $C := (x^1, \ldots, x^r, x^1)$ be a trap cycle in the state transition graph $\mathcal{S}_N$. Let $I$ be the set of components $i$ such that there exists $j_1, j_2$ satisfying $x_i^{j_1} \neq x_i^{j_2}$. Then there is a singular steady state such that $s_i = \theta$ for all $i \in I$.*

The proofs in [7] show how to derive singular steady states satisfying the statements of Prop. 4.8 and 4.9. However, those singular steady states may coincide with $(\theta, \ldots, \theta)$, even when $I \neq \{1, \ldots, n\}$.

# 5 Perspectives

We have seen in this paper that it is possible to relate systematically singular steady states to attractors in the state transition graph. To do so, we often exploit knowledge about the structure of the associated interaction graph. The results obtained illustrate the possibilities of studying the dynamical behavior of the system without the explicit use of the state transition graph. However, we have focussed on a coarse description, characterizing state transition graphs by the number of their attractors, and distinguishing attractors by their cardinality. In order to tap the full potential of this approach to analyzing the system's dynamics, it should be refined

further. A promising starting point for future work is the concept of local interaction graphs introduced in [3]. The authors associate every state of the system with an interaction graph, the union of which is the global interaction graph. This approach allows for a better understanding of what structures in the interaction graph influence the system's behavior in a given state. Combining this local view with our understanding of singular steady states may yield a more detailed description of the resulting dynamical behavior.

## Acknowledgment

We would like to thank an anonymous referee for suggestions to simplify several proofs in an earlier version of the paper.

## References

[1] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. Application of formal methods to biological regulatory networks: extending Thomas' asynchronous logical approach with temporal logic. *J. Theoret. Biol.*, 229:339–347, 2004.

[2] E. Remy, B. Mossé, C. Chaouiya, and D. Thieffry. A description of dynamical graphs associated to elementary regulatory circuits. *Bioinformatics*, 19:172–178, 2003.

[3] E. Remy, P. Ruet, and D. Thieffry. Graphic requirements for multistability and attractive cycles in a boolean dynamical framework. Prépublication, 2005.

[4] E. Remy, P. Ruet, and D. Thieffry. Positive or negative regulatory circuit inference from multilevel dynamics. In *Positive Systems: Theory and Applications*, volume 341 of *LNCIS*, pages 263–270. Springer, 2006.

[5] A. Richard and J.-P. Comet. Necessary conditions for multistationarity in discrete dynamical systems. Rapport de Recherche, 2005.

[6] A. Richard, J.-P. Comet, and G.Bernot. R. Thomas' modeling of biological regulatory networks: introduction of singular states in the qualitative dynamics. *Fund. Inform.*, 65:373–392, 2005.

[7] H. Siebert and A. Bockmayr. Relating attractors and singular steady states in the logical analysis of bioregulatory networks. Preprint 373, DFG Research Center Matheon, 2007.

[8] E. H. Snoussi and R. Thomas. Logical identification of all steady states: the concept of feedback loop characteristic states. *Bull. Math. Biol.*, 55:973–991, 1993.

[9] C. Soulé. Graphical requirements for multistationarity. *ComPlexUs*, 1:123–133, 2003.

[10] R. Thomas. Boolean formalization of genetic control circuits. *J. Theoret. Biol.*, 42:563–585, 1973.

[11] R. Thomas and R. d'Ari. *Biological Feedback*. CRC Press, 1990.

[12] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. II. Logical analysis of regulatory networks in terms of feedback circuits. *Chaos*, 11:180–195, 2001.

# Chapter 3

# Deriving Behavior of Boolean Bioregulatory Networks from Subnetwork Dynamics

Heike Siebert.

*Mathematics in Computer Science*, 2(3):421-442, 2009.

**Remarks.** Some core results of the preceding chapter are generalized to arbitrary Boolean functions representing regulatory networks. Singular steady states are now used to construct the state transition graph of a network from the dynamics of suitable subnetworks. I introduce a procedure for deriving singular steady states using constraint propagation via an iteration process. This allows for particularly efficient and comprehensive treatment of networks with input layer, which are often used, e.g., for modeling signaling networks.

# Deriving Behavior of Boolean Bioregulatory Networks from Subnetwork Dynamics

Heike Siebert

*Abstract.* In the well-known discrete modeling framework developed by R. Thomas, the structure of a biological regulatory network is captured in an interaction graph, which, together with a set of Boolean parameters, gives rise to a state transition graph describing all possible dynamical behaviors. For complex networks the analysis of the dynamics becomes more and more difficult, and efficient methods to carry out the analysis are needed. In this paper, we focus on identifying subnetworks of the system that govern the behavior of the system as a whole. We present methods to derive trajectories and attractors of the network from the dynamics suitable subnetworks display in isolation. In addition, we use these ideas to link the existence of certain structural motifs, namely circuits, in the interaction graph to the character and number of attractors in the state transition graph, generalizing and refining results presented in [10]. Lastly, we show for a specific class of networks that all possible asymptotic behaviors of networks in that class can be derived from the dynamics of easily identifiable subnetworks.

## 1 Introduction

When modeling biological systems, one first has to decide what kind of modeling framework is best suited to incorporate the available data and to yield results without too many additional assumptions about the system. If only coarse information is available, as is often the case when studying biological regulatory networks, logical modeling approaches lend themselves well to capturing the essential, qualitative features of the system. In the 70's, R. Thomas introduced a discrete modeling formalism, which has been continuously further developed and successfully applied to biological problems (see [13], [14] and references therein). The structure of the network is captured in a directed, signed graph called interaction graph. Edges represent activating or inhibiting interactions between components, which in turn are represented by Boolean variables. A component is considered active if the associated variable has value 1 and inactive otherwise. Boolean parameter values specify a function that determines the dynamical behavior. Biologically realistic rules are employed to derive a state transition graph from the Boolean function, which amounts to a non-deterministic representation of all possible behaviors of the system.

Since the representation of the dynamics is in some sense comprehensive, the analysis of the behavior is rather involved for complex networks. One idea to simplify the analysis is to deconstruct the complex network into simpler building blocks, to analyze their dynamics in isolation and then derive information about the network dynamics from the subnetwork

behavior. Clearly, this method will not work for arbitrarily chosen subnetworks, because further components and interactions influence its behavior once it is embedded in a complex system. If, however, we can identify subnetworks that somehow govern the behavior of the larger network, at least in some part of state space, then we can infer useful results about the whole network from looking at those subnetworks.

We approach the problem of finding suitable subnetworks in the following way. We basically look for parts of state space, where some of the components of the network remain stable independent of the values of other network components. A notion that formalizes this idea is the notion of singular steady state. It was first introduced by R. Thomas and E.H. Snoussi in [11] for a certain class of multi-valued discrete functions. Here, a so-called *singular value* represent the threshold of an interaction, which allows a refined representation of the network dynamics. We adapted these ideas to a Boolean setting in [10], resulting in definitions and results for a restricted class of Boolean models. In this paper, we substantially generalize the framework introduced in [10]. We allow characteristics, i. e., the sign of network interactions to depend on the current state of the system. Whether a component has an activating or inhibiting influence on its target may depend on the activity of certain cofactors. A well-known example is the DNA-binding protein TCF which can repress as well as activate the same target genes. TCF acts as activator in the presence of $\beta$-catenin, induced by WNT signaling, while the co-expression of the protein TLE converts TCF into a repressor. We call systems including such ambiguous interactions *context sensitive*. Adaptations in the definition of interaction graphs and parameters allow us to include context sensitive systems in our considerations.

In this setting, a component can adopt the *singular value* $\theta$ in addition to the two *regular values* 1 and 0. Again, the singular value can be interpreted as the threshold value of an interaction, and thus represents a state where we do not know whether or not the corresponding interaction is active. Considering such states can help us to a clearer understanding of the component dependencies in the system. To obtain a refined representation of the structure of the system with respect to its dynamics, we exploit the concept of local interactions graphs. It was already successfully used in [5] and [4], and allows for a better understanding of what structures in the interaction graph influence the system's behavior in a given state. Combining these ideas, we introduce local interaction graphs of singular steady states which allow us to identify subnetworks that govern the behavior of the whole system.

The paper is organized as follows. In Sect. 2 we introduce the Boolean framework we use to describe regulatory networks. The following section clarifies the relation between the function $f$ governing the system's dynamics and the structure of the interaction graph, which leads to the notion of local interaction graph. Subsequently, we introduce singular steady states. In Sect. 5, we employ the concept of local interaction graphs for singular steady states. We identify subnetworks that govern the system's behavior, and introduce a procedure to derive the dynamics and in particular attractors of the network from the corresponding subnetwork dynamics. We also focus on subnetworks that basically consist of an isolated circuit, and link there existence in the interaction graph to certain dynamical characteristics of the network. In Sect. 6, we then analyze a specific class of networks the properties of which allow us to easily find subnetworks from which we can derive all possible attractors of the original system. We end the paper with concluding remarks and perspectives for future work.

This is the extended version of a paper presented at the Algebraic Biology conference 2008 [9].

## 2 Regulatory Networks

As already mentioned, a directed, signed graph is used in the Thomas formalism to capture the network structure of a regulatory system. We are now interested in a more general representation that allows for the interaction sign to depend on the current state of the system. To accurately describe the structure of such context sensitive networks we use directed multigraphs that allow for parallel edges. Multigraphs have been used in a similar way in [2]. We set $\mathcal{B} := \{0, 1\}$.

**Definition 2.1.** *An* interaction (multi-)graph *(or bioregulatory (multi-)graph) $\mathcal{I}$ is a labeled directed multigraph with vertex set $V := \{\alpha_1, \ldots, \alpha_n\}$, $n \in \mathbb{N}$, and edge set $E \subseteq V \times V \times \{+, -\}$.*

The vertices $\alpha_1, \ldots, \alpha_n$ represent the components of the regulatory network such as genes, RNA, or proteins. We view each component $\alpha_i$ as a variable that adopts values in $\mathcal{B}$. The value 1 signifies that the component is active, i. e., it influences its interaction targets according to the interaction signs. For example, if some substance concentration needs to cross a threshold in order to influence some target component, then the corresponding Boolean value is 0 as long as the concentration is below, and 1 if the concentration is above the threshold.

When analyzing the interaction graph of a network we are interested in certain structural motives. We focus on so-called *(feedback) circuits*. Here, a circuit is a tuple $(e_1, \ldots, e_r)$ of edges $e_i = (k^i, l^i, \varepsilon) \in E$ such that $l^i = k^{i+1}$ for all $i \in \{1, \ldots, r\}$ modulo $r$, and all $k^i$ (and therefore $l^i$), $i \in \{1, \ldots, r\}$, are pairwise distinct. The *sign of a circuit* is the product of the signs of its edges. Note that in a multigraph a circuit is not uniquely determined by its vertices. Figure 1 shows an interaction graph with two circuits consisting of the vertices $\alpha_2$ and $\alpha_3$: the positive circuit $((\alpha_2, \alpha_3, +), (\alpha_3, \alpha_2, +))$ and the negative circuit $((\alpha_2, \alpha_3, +), (\alpha_3, \alpha_2, -))$.

To simplify notation, we identify each vertex $\alpha_i$ with its index $i$, and denote $e_{ij}^{\varepsilon} := (i, j, \varepsilon)$ for all $(i, j, \varepsilon) \in E$. For each $\alpha_i$ we denote by $Pred(\alpha_i)$ the set of *predecessors* of $\alpha_i$, i. e., the set of vertices $\alpha_j$ such that there is an edge $(\alpha_j, \alpha_i, \varepsilon)$ for some $\varepsilon \in \{+, -\}$ in $E$. To identify parallel edges we set $E'' = \{(i, j) \mid \exists e_{i,j}^{\varepsilon}, e_{i,j}^{\varepsilon'} \in E : \varepsilon \neq \varepsilon'\}$ and $E' = E \setminus E''$.

The information inherent in the interaction graph is in general not sufficient to determine the specific dynamical behavior of a system. Next we give a formal definition of the term *bioregulatory network* that includes both the structure of the network and the rules governing its dynamics. The notation is based on ideas introduced in [1] and [7].

**Definition 2.2.** *Let $\mathcal{I} = (V, E)$ be an interaction graph comprising $n$ vertices. A* state *of the system described by $\mathcal{I}$ is a tuple $s \in \mathcal{B}^n$. The set of (regular)* resource edges *$R_j(s) = R_j^{\mathcal{I}}(s)$ of $\alpha_j$ in state $s = (s_1, \ldots, s_n)$ is the set*

$$\{(\alpha_i, \alpha_j, \varepsilon) \in E \mid (\varepsilon = + \wedge s_i = 1) \vee (\varepsilon = - \wedge s_i = 0)\}.$$

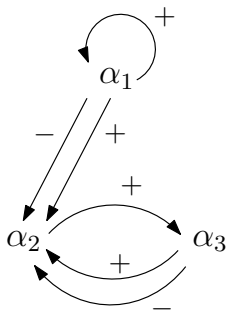
*Given a set*

$$K(\mathcal{I}) := \{K_{j,R_j(s)} \mid j \in \{1, \ldots, n\}, s \in \mathcal{B}^n\}$$

*of* (logical) parameters*, which adopt values in $\mathcal{B}$, we define the Boolean function*

$$f = f^{K(\mathcal{I})} : \mathcal{B}^n \to \mathcal{B}^n, \; s \mapsto (K_{1,R_1(s)}, \ldots, K_{n,R_n(s)}) .$$

*The pair $N := (\mathcal{I}, f)$ is called* bioregulatory network.




$$
\begin{array}{lcl}
K_{1,\emptyset} & = & 0 \\
K_{1,\{e^+_{11}\}} & = & 1 \\
K_{2,\{e^+_{12},e^+_{32}\}} & = & 0 \\
K_{2,\{e^+_{12},e^-_{32}\}} & = & 1 \\
K_{2,\{e^-_{12},e^+_{32}\}} & = & 1 \\
K_{2,\{e^-_{12},e^-_{32}\}} & = & 0 \\
K_{3,\emptyset} & = & 0 \\
K_{3,\{e^+_{23}\}} & = & 1
\end{array}
$$

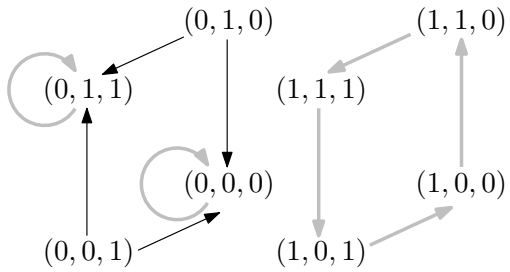



Figure 1: Interaction graph of a system comprising three components, a list of all parameters with an assignment of Boolean values, and the corresponding state transition graph. The heavier gray edges indicate attractors.

The behavior of a component $\alpha_j$ is determined by the influences its predecessors exert on it. The set of resource edges $R_j(s)$ contains all edges that contribute to an activation of $\alpha_j$ in state $s$. Note that here the absence of an inhibiting influence (represented by a negative edge) is interpreted as an activating influence on the target component. With this interpretation we have that for every $s \in \mathcal{B}^n$ there is $\varepsilon \in \{+,-\}$ such that $e^\varepsilon_{ij} \in R_j(s)$, if $(i,j) \in E''$. If $(i,j) \in E'$, then $R_j(s)$ may or may not contain the corresponding edge $e^\varepsilon_{i,j}$, depending on $s$.

In Fig. 1 an interaction graph and a choice of parameter values are given. For $\alpha_1$ and $\alpha_3$ the parameters depend on whether or not the single positive edge ending in $\alpha_1$ resp. $\alpha_3$ is effective or ineffective. We have $R_1(s) = \emptyset$ for all states $s$ with $s_1 = 0$, and $R_1(s) = \{e^+_{11}\}$ for all $s$ with $s_1 = 1$, while the resource edge sets for $\alpha_3$ depend similarly on $e^+_{23}$. The component $\alpha_2$ is influenced by both $\alpha_1$ and $\alpha_3$ via two parallel edges, respectively. Thus the set of resources is never empty. For example, we have $R_2((0,0,1)) = \{e^-_{12}, e^+_{32}\}$. A closer look at the choice of parameter values allows the following interpretation. If $\alpha_1$ has activity level 0, then the influence of $\alpha_3$ on $\alpha_2$ corresponds to an activating influence: if $\alpha_3$ is inactive, $\alpha_2$ tends to inactivity represented by the parameter $K_{2,\{e^-_{12},e^-_{32}\}} = 0$, and if $\alpha_3$ is active $\alpha_2$ tends to activity since $K_{2,\{e^-_{12},e^+_{32}\}} = 1$. If $\alpha_1$ has value 1, then the situation is reversed and $\alpha_3$ inhibits $\alpha_2$. The system is context sensitive.

In [10], the parameters correspond to sets of resource vertices, i. e., the influence of one component on another cannot change depending on the current state of the system. The network shown in Fig. 1 cannot be represented with that restriction. However, the notion of resource edges and resource vertices are equivalent, if there are no parallel edges in the interaction graph.

The parameters determine the behavior of the system as follows. The Boolean value of the parameter $K_{j,R_j(s)}$ indicates how the activity level, i. e., the value of the component $\alpha_j$ will evolve from its value in state $s$. It will increase (resp. decrease) if the parameter value is greater (resp. smaller) than $s_i$. The activity level stays the same if both values are equal. Thus, the function $f$ maps a state $s$ to the state the system tends to evolve to. However, if a state and its image differ in more than one component, we take the following consideration into account. In a biological system two different processes of change in activity level represented by the value change of two distinct components will not take the exact same amount of time. Thus we assume

that in the discrete dynamical representation a state differs from its successor in at most one component. This procedure is called *asynchronous update* in Thomas' framework. By applying this idea we derive a non-deterministic representation of the dynamics which we again formalize as a directed graph.

**Definition 2.3.** *The* state transition graph $\mathcal{S}_N$ *describing the dynamics of the network $N$ is a directed graph with vertex set $\mathcal{B}^n$. For states $s = (s_1, \dots, s_n)$ and $s' = (s'_1, \dots, s'_n)$, there is an edge $s \to s'$ if and only if $s' = f(s) = s$ or $s'_i = f_i(s)$ for some $i \in \{1, \dots, n\}$ satisfying $s_i \neq f_i(s)$ and $s'_j = s_j$ for all $j \neq i$.*

On the right in Fig. 1 we see the state transition graph corresponding to the given interaction graph and parameters. The dynamics are non-deterministic. For example, there are two edges leaving the state (0,1,0), representing two different behaviors of the system.

Every possible behavior of the system is captured in the corresponding state transition graph. To analyze the graph we use, in addition to standard terminology from graph theory such as paths and cycles, the following concepts.

**Definition 2.4.** *An infinite path $(s_0, s_1, \dots)$ in $\mathcal{S}_N$ is called* trajectory*. A nonempty set of states $D$ is called* trap set *if every trajectory starting in $D$ never leaves $D$. A trap set $A$ is called* attractor *if for all $s^1, s^2 \in A$ there is a path from $s^1$ to $s^2$ in $\mathcal{S}_N$. A cycle $C := (s^1, \dots, s^r, s^1)$, $r \geq 2$, is called a* trap cycle *if every $s^j$, $j \in \{1, \dots, r\}$, has only one outgoing edge in $\mathcal{S}_N$, i. e., the trajectory starting in $s^1$ is unique. A state $s$ is called* steady state*, if there exists an edge $s \to s$, i. e. if $f(s) = s$.*

In other words, the attractors correspond to the terminal strongly connected components of the graph. Steady states as well as trap cycles are attractors. The attractors in the state transition graph in Fig. 1 are the sets containing the steady states, i. e., $\{(0,0,0)\}$ and $\{(0,1,1)\}$, and the set containing the states of the trap cycle in the graph, i. e., $\{(1,0,0),(1,1,0),(1,1,1),(1,0,0)\}$.

The behavior of a system becomes, at least to some degree, predictable and stable inside an attractor. Often, a sensible biological interpretation can be found for an attractor. In cell differentiation, the different stable states reached at the end of development may be represented by distinct steady states in the state transition graph. Attractors of cardinality greater than one imply cyclic behavior, and thus can often be identified with homeostasis of sustained oscillatory activity, as can be found in the cell cycle or circadian rhythm.

State transition graphs always contain at least one attractor. The proof of the following more precise statement can be found in [10].

**Proposition 2.5.** *For every state $s \in \mathcal{B}^n$ exists a trajectory in $\mathcal{S}_N$ which starts in $s$ and leads to an attractor.*

Coming to the end of this section, we note a useful observation. If some vertex $\alpha_i$ in $\mathcal{I}$ does not have a predecessor, then clearly $a_i = K_{i,\emptyset}$ for every state $a = (a_1, \dots, a_n)$ in an attractor. Similarly, we know the values $a_j$ for vertices the only predecessor of which is $\alpha_i$, and so on. That is, we can easily determine the dynamical behavior of such vertices, which leads to the same fixed values of those components for every initial state of the system. Throughout the remainder of the paper we exclude such components and assume that every vertex in $\mathcal{I}$ has a predecessor. We still allow the system to have input values in the sense of components maintaining their current activity level independent of the values of the other components. Such an input component is represented as a vertex with its only incoming edge being a positive self-loop.

## 3 Functionality and Local Interaction Graphs

Throughout the paper let $N := (\mathcal{I} = (V, E), f = f^{K(\mathcal{I})})$ be a bioregulatory network comprising $n$ components. When analyzing the system, it is an interesting question whether it is possible to link structural network characteristics to dynamical characteristics. In order to obtain sensible results, however, we need to make sure that the structure of the network captured in the interaction graph and the rules governing the behavior of the system represented by the Boolean function $f$ do not contradict each other. That is, the choice of parameter values should be consistent with the information inherent in the interaction graph. We require that each edge represented in the interaction graph should have a notable effect on the system's dynamics. Moreover, the edge's character given by its sign should be reflected in its dynamical impact.

To formalize these requirements we introduce the following notation. Recall that we interpret the absence of inhibition as a potentially activating effect. So, if we deal with parallel edges from a vertex $i$ to a vertex $j$, that is if $(i, j) \in E''$, we know that for every $s \in \mathcal{B}^n$ there is $\varepsilon \in \{+, -\}$ such that $e^{\varepsilon}_{ij} \in R_j(s)$. If there is only a single edge, i. e. $(i, j) \in E'$, then $R_j(s)$ may or may not contain the corresponding edge $e^{\varepsilon}_{i,j}$, depending on $s$.

For $j \in \{1, \ldots, n\}$, set $M^{\mathcal{I}}_j := M_j := \{R_j(s) \mid s \in \mathcal{B}^n\}$. Then, by the above considerations, each $M \in M_j$ can be written as $M = \bigcup_{i \in Pred(j)} L_i$ with $L_i = \{e^{\varepsilon}_{i,j}\}$ for some $\varepsilon \in \{+, -\}$, if $(i, j) \in E''$, and $L_i = \emptyset$ or $L_i = \{e^{\varepsilon}_{i,j}\} \subset E$, if $(i, j) \in E'$. By definition we have

$$K(\mathcal{I}) = \{K_{j,M} \mid j \in \{1, \ldots, n\},\, M \in M_j\}\,.$$

We want to ensure that the choice of parameter values does not contradict the information inherent in the interaction graph. As mentioned above, that means we have to check for the existence and the character, positive or negative, of influence on the system dynamics exhibited by each interaction. To do so, we again have to distinguish between edges in $E'$ and $E''$. For $e = (i, j, \varepsilon)$, $(i, j) \in E'$, we have $M \cup \{e\} \in M_j$ for all $M \in M_j$, and we demand that $K_{j,M} \leq K_{j,M\cup\{e\}}$ for all $M \in M_j$. Recall that the addition of an edge to the set of resources always signifies increasing activating influence. So, the condition ensures that increasing activating influence does not result in a decrease of component activity level. To ensure that $e$, at least for some state, has a notable impact on the dynamics, we extend the condition and get:

$$\forall\, M \in M_j \,:\, K_{j,M} \leq K_{j,M\cup\{e\}} \quad \text{and} \quad \exists\, M' \in M_j \,:\, K_{j,M'} < K_{j,M'\cup\{e\}}\,. \tag{1}$$

In the case $(i, j) \in E''$, there exists $e' = (i, j, \varepsilon')$ with $\varepsilon \neq \varepsilon'$. Since $\alpha_i$ influences $\alpha_j$ positively as well as negatively depending on the current state, we cannot impose a general monotonicity condition on the parameters as in the first part of (1). However, again we require that there is at least one state where the addition of $e$ to the set of resources induces an increase in the parameter value. Otherwise the edge $e$ would be superfluous. Since in every given state either $e$ or $e'$ is contained in the set of resources, we compare parameter values for sets $M \in M_j$ and $(M \setminus \{e'\}) \cup \{e\}$. We obtain the condition

$$\exists\, M' \in M_j \,:\, K_{j,M'} < K_{j,(M'\setminus\{e'\})\cup\{e\}}\,. \tag{2}$$

We call edges that satisfy condition (1) resp. (2) *functional*. This concept of functionality is an adaptation of the notion of functionality introduced in [10]. In the following, we always assume that all edges in the interaction graph are functional.

We have already seen in Sect. 2 that for the example in Fig. 1 $R_1(s) = \emptyset$ for all states $s$ with $s_1 = 0$, and $R_1(s) = \{e_{11}^+\}$ for all $s$ with $s_1 = 1$. Thus $M_1 = \{\emptyset, \{e_{11}^+\}\}$. Similarly $M_3 = \{\emptyset, \{e_{23}^+\}\}$. The choice of Boolean values for the parameters satisfies condition (1) and ensures the functionality of the edges $e_{11}^+$ and $e_{23}^+$. Since $\alpha_2$ is influenced by both $\alpha_1$ and $\alpha_3$ via two parallel edges, we get $M_2 = \{\{e_{12}^+, e_{32}^+\}, \{e_{12}^+, e_{32}^-\}, \{e_{12}^-, e_{32}^+\}, \{e_{12}^-, e_{32}^-\}\}$. Again the choice of parameter values renders all edges functional.

A different choice of parameter values leads to different results. If we set $K_{2,\{e_{12}^+, e_{32}^+\}} = K_{2,\{e_{12}^-, e_{32}^+\}} = 1$ and $K_{2,\{e_{12}^+, e_{32}^-\}} = K_{2,\{e_{12}^-, e_{32}^-\}} = 0$, then verification of conditions (1) and (2) shows that $e_{12}^+$, $e_{12}^-$ and $e_{32}^-$ are not functional. Only the edge $e_{32}^+$ is functional and influences the system's dynamics.

It now may seem that we put a lot of restrictions on the choice of parameter values and thus on the function $f$ by demanding functionality of all edges. But in fact we only make sure that the graph representation of the system's structure fits the structural information encoded in $f$. This is no restriction on the Boolean function $f$ as the following statement shows. The proof can be found in [9].

**Proposition 3.1.** *Let $g : \mathcal{B}^n \to \mathcal{B}^n$ be a Boolean function. Then there exists an interaction graph $\mathcal{I} = (V, E)$ and a set of parameters $K(\mathcal{I})$ such that $g = f^{K(\mathcal{I})}$.*

The above statement also illustrates the fact that the interaction graph holds only coarse information on the system. The same interaction graph may give rise to different dynamics depending on the choice of parameter values. However, a more refined understanding of the network structure is possible, if we consider the impact of interactions on the dynamics with respect to the current state of the system.

Since all edges in the interaction graph are functional, we know that each edge has an impact on the dynamics. However, this influence does not have to be effective in the whole state space $\mathcal{B}^n$. To capture local structural aspects we introduce the concept of *local interaction graphs*. It has already been used in [5] and [4] (see also references therein). In the following, we denote with $\overline{s}^i$ the state that coincides with $s$ in all components $j \neq i$ and takes the value $1 - s_i$ in the $i$-th component.

**Definition 3.2.** *Let $\mathcal{I} = (V, E)$ be an interaction graph with parameter set $K(\mathcal{I})$. Let $s = (s_1, \ldots, s_n) \in \mathcal{B}^n$. Then we denote by $\mathcal{I}(s)$ the graph with vertex set $V$ and edge set $E(s) \subseteq E$. An edge $(i, j, \varepsilon)$ is in $E(s)$ if and only if*

$$K_{j,R_j(s)} \neq K_{j,R_j(\overline{s}^i)} \quad \wedge \quad \varepsilon = + \Leftrightarrow s_i = K_{j,R_j(s)}\,.$$

*We call $\mathcal{I}(s)$ the* (local) interaction graph in state $s$.

Clearly, every edge in the local interaction graph $\mathcal{I}(s)$ is also contained in $\mathcal{I}$, since we use the same parameters to characterize the edges. More precisely, $\mathcal{I}$ is the union of all graphs $\mathcal{I}(s)$, $s \in \mathcal{B}^n$. We call $\mathcal{I}$ also the *global interaction graph*. Note that there are no parallel edges in a local interaction graph. Figure 2(a) and (b) show the graphs $\mathcal{I}((0,0,0))$ and $\mathcal{I}((1,0,0))$ corresponding to the example given in Fig. 1. The local interaction graphs give us a finer understanding of the way the network components interact. They can be seen as a visualization of the discrete Jacobian matrix of the Boolean function $f^{\mathcal{I}} = f$ as introduced in [8], since we have $f_j(s) = K_{j,R_j(s)}$ for all $s \in \mathcal{B}^n$.

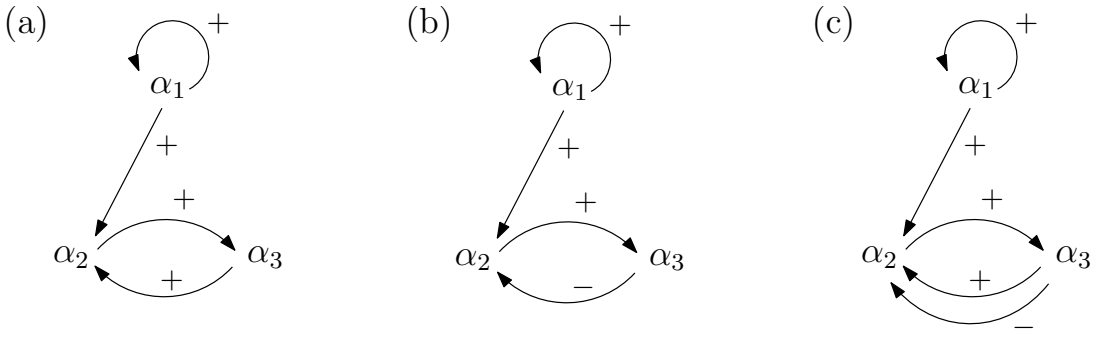



Figure 2: Local interaction graphs corresponding to the graph and parameters given in Fig. 1. $\mathcal{I}((0,0,0))$ in (a), $\mathcal{I}((1,0,0))$ in (b), $\mathcal{I}((\theta,0,0))$ in (c).

# 4 Singular States

In our formalism we only consider whether a component is active or not. We now incorporate a threshold value that allows us to express uncertainty in the sense that we do not know if a certain interaction is effective. We already used this concept in [10] for networks without context sensitivity. Again, we mainly use notation introduced in [7].

**Definition 4.1.** *Set $\mathcal{B}_\theta := \{0, \theta, 1\}$, where $\theta$ is a symbolic representation of the threshold value and satisfies the order $0 < \theta < 1$. We allow each regulatory component $\alpha_i$ to take values in $\mathcal{B}_\theta$. The values 0 and 1 are called* regular values *and $\theta$ is called* singular value. *The elements of $\mathcal{B}_\theta^n$ are called* states. *If all components of a state are regular, it is called* regular state, *else it is called* singular state. *For every state $s = (s_1, \ldots, s_n)$ we define $J(s) := \{i \in \{1, \ldots, n\} \mid s_i = \theta\}$.*

We call $|a, b|$ a *qualitative value* if $a, b \in \mathcal{B}$ and $a \leq b$. The qualitative value $|0, 0|$ is identified with the regular value 0, $|1, 1|$ with the regular value 1, and $|0, 1|$ with the singular value $\theta$. The relations $<$, $>$, and $=$ are used with respect to this identification.

In the following, we denote $[s] := \{s' \in \mathcal{B}^n \mid s'_j = s_j \text{ for all } j \notin J(s)\}$ for all $s \in \mathcal{B}_\theta^n$. The set $[s]$ describes a part of the regular state space which we can be projected on $\mathcal{B}^k$, where $k$ is the cardinality of $J(s)$.

**Definition 4.2.** *We define for all $i \in \{1, \ldots, n\}$*

$$f^\theta = f^{K(\mathcal{I}),\theta} : \mathcal{B}_\theta^n \to \mathcal{B}_\theta^n \quad \text{by} \quad f_i^\theta(s) = |K_{i,\min(s)}, K_{i,\max(s)}|\,,$$

*where $K_{i,\min(s)} := \min\{ K_{i,R_i(s')} \mid s' \in [s]\}$ and $K_{i,\max(s)} := \max\{ K_{i,R_i(s')} \mid s' \in [s]\}$. We call $s \in \mathcal{B}_\theta^n$ a* steady state *if $f^\theta(s) = s$.*

The definition of $K_{i,\min(s)}$ and $K_{i,\max(s)}$ ensures that the image of a regular state under $f^\theta$ is again a regular state. More specific, we have $f^\theta|_{\mathcal{B}^n} = f$. If a state has singular components, then $K_{i,\min(s)}$ and $K_{i,\max(s)}$ compare the best and worst case scenario regarding activation for component $i$ by considering all possible combinations of regular values for the singular components of $s$. If the value $f_i^\theta(s)$ for some $s \in \mathcal{B}_\theta^n$ is regular, we can deduce that the regular components of $s$ already determine the behavior of the $i$-th component, independent of any predecessors with singular value in $s$. If it is not regular, then we simply do not have enough information to

predict what its future value would be. A singular steady state can then be viewed as a partial steady state of the regular dynamics. We will exploit this fact in the next section.

Thomas and Snoussi already link singular states to circuits in the interaction graph, albeit in a different framework (see [11]). We have adapted their ideas to a Boolean framework without context sensitivity in [10].

**Definition 4.3.** *Let $C = (\alpha_{i_1}, \ldots, \alpha_{i_r})$ be a circuit in $\mathcal{I}$. A state $s = (s_1, \ldots, s_n) \in \mathcal{B}_\theta^n$ is called* characteristic state *of $C$ if $s_{i_l} = \theta$ for all $l \in \{1, \ldots, r\}$.*

In general, a characteristic state of a circuit is not unique. The state $(\theta, \ldots, \theta)$ is characteristic for every circuit in $\mathcal{I}$. A simple modification of the reasoning in [10] leads to the following statement.

**Theorem 4.4.** *Every singular steady state is characteristic of some circuit in $\mathcal{I}$.*

A singular steady state $s$ can be characterized using only regular states and the function $f$. The idea is to check component-wise the behavior for regular states $s^+$ and $s^-$ that satisfy $K_{i,R_i(s^+)} = K_{i,\max(s)}$ and $K_{i,R_i(s^-)} = K_{i,\min(s)}$ for some $i \in \{1, \ldots, n\}$. The proofs for networks that are not context sensitive are given in [10] and can be easily adapted.

# 5 Subnetworks Governing Network Behavior

In this section we try to find subnetworks of $N$ that in some sense govern the behavior of the whole system, at least in some part of state space. The key idea in this endeavor is to have a closer look at the structural as well as dynamical information inherent in a given singular steady state. As a first step, we adapt the concept of local interaction graphs to singular states.

Recall that $J(s)$ is the set of all singular components of a state $s \in \mathcal{B}_\theta^n$, and $[s] := \{s' \in \mathcal{B}^n \mid s'_j = s_j \text{ for all } j \notin J(s)\}$.

**Definition 5.1.** *Let $s = (s_1, \ldots, s_n) \in \mathcal{B}_\theta^n$. We denote by $\mathcal{I}(s)$ the (multi-)graph with vertex set $V$ and edge set $E(s)$. An edge $e$ is in $E(s)$ if and only if there exists $s' = (s'_1, \ldots, s'_n) \in [s]$ such that $e \in E(s')$, where $E(s')$ denotes the edge set of the interaction graph $\mathcal{I}(s')$ in $s'$. Again, we call $\mathcal{I}(s)$ the* (local) interaction graph in $s$.

Note that the interaction graph in a singular state may have parallel edges. In Fig. 2 (c) we see the local interaction graph in state $(\theta, 0, 0)$, which is the union of the graphs $\mathcal{I}((0, 0, 0))$ and $\mathcal{I}((1, 0, 0))$ given in (a) and (b).

A singular steady state $s$ yields stability in the dynamical behavior for the components that do not belong to $J(s)$. To make a more precise statement we introduce notation for a specific subgraph of $\mathcal{I}(s)$. By $\mathcal{I}^\theta(s)$ we denote the (multi-)graph with vertex set $V^\theta(s) := J(s)$ and edge set $E^\theta(s) := \{(i, j, \varepsilon) \in E(s) \mid i, j \in J(s)\}$. That is, we only keep the singular components and interactions between them. We call a graph $Z$ component of $\mathcal{I}^\theta(s)$, if $Z = (V_Z, E_Z)$ is a maximal subgraph of $\mathcal{I}^\theta(s)$ such that for every $k, k' \in V_Z$ exist vertices $k_1, \ldots, k_r \in V_Z$ with $k_1 = k$, $k_r = k'$, and $(k_i, k_{i+1}, \varepsilon) \in E^\theta(s)$ or $(k_{i+1}, k_i, \varepsilon) \in E^\theta(s)$ for some $\varepsilon \in \{+, -\}$ and all $i \in \{1, \ldots, r-1\}$. In Fig. 3 we see for our running example introduced in Fig. 1 the graphs $\mathcal{I}((0, \theta, \theta))$ and $\mathcal{I}^\theta((0, \theta, \theta))$ in (a), as well as the graphs $\mathcal{I}((1, \theta, \theta))$ and $\mathcal{I}^\theta((1, \theta, \theta))$ in (b). Lastly, let $C$ be a circuit in $\mathcal{I}(s)$ such that all edges of $C$ are in $\mathcal{I}^\theta(s)$. Then there exists a component of $\mathcal{I}^\theta(s)$ that contains $C$. We denote this component by $J_C(s)$. The next lemma shows that the

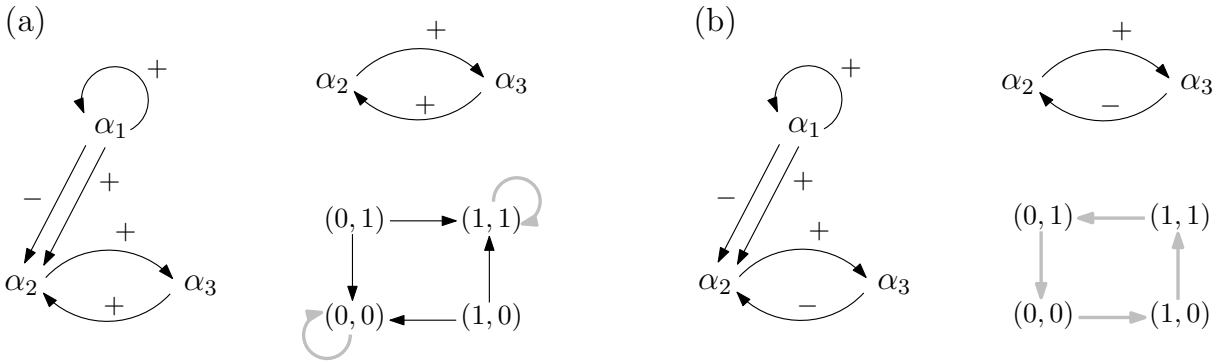



Figure 3: Local interaction graph $\mathcal{I}((0,\theta,\theta))$ on the left, $\mathcal{I}^\theta((0,\theta,\theta))$ and corresponding state transition graph on the right of (a). Local interaction graph $\mathcal{I}((1,\theta,\theta))$ on the left, $\mathcal{I}^\theta((1,\theta,\theta))$ and corresponding state transition graph on the right of (b). Attractors are indicated by heavier gray edges.

stability of the regular components of a singular steady state is not influenced by value changes in a component $Z$ of $\mathcal{I}^\theta(s)$. Moreover, if $\mathcal{I}^\theta(s)$ has more than one component, the component dynamics are independent of each other. This property is crucial for the remaining results in this section. The proof of the lemma is an adaptation of a similar, less general statement in [10]. Note that in [10] a different definition of $\mathcal{I}^\theta(s)$ is used that does not take the effectiveness of interactions in state $s$ into account.

**Lemma 5.2.** *Let $s = (s_1, \ldots, s_n)$ be a singular steady state, and let $Z_1, \ldots, Z_m$ be the components of $\mathcal{I}^\theta(s)$. Consider a union $Z$ of arbitrary components $Z_j$. Let $\tilde{s} = (\tilde{s}_1, \ldots, \tilde{s}_n) \in \mathcal{B}_\theta^n$ such that $\tilde{s}_i = s_i$ for all $i \notin Z$. Then $f_i^\theta(\tilde{s}) = f_i^\theta(s) = s_i = \tilde{s}_i$ for all $i \notin Z$.*

*Proof.* First, let us consider $i \notin J(s)$. Since $s_j = \theta$ for all $j \in Z$, we have $J(\tilde{s}) \subseteq J(s)$. Therefore, $[\tilde{s}] \subseteq [s]$. It follows that $K_{i,\min(s)} \leq K_{i,\min(\tilde{s})} \leq K_{i,\max(\tilde{s})} \leq K_{i,\max(s)}$. Since $f_i^\theta(s) = s_i$ is regular, we know $K_{i,\min(s)} = K_{i,\max(s)} = s_i$. Thus, $K_{i,\min(\tilde{s})} = K_{i,\max(\tilde{s})} = s_i$ and $f_i^\theta(\tilde{s}) = s_i = \tilde{s}_i$.

Now, let us consider $i \in J(s) \setminus Z$. Then there is no vertex $\alpha_j$ in $Z$ that is a predecessor of $\alpha_i$ in the local interaction graph $\mathcal{I}(s)$. So there is no interaction from a component in $Z$ to $\alpha_i$ that is functional in the part of state space given by $[s]$. Since $[\tilde{s}] \subseteq [s]$, changes in the values of components in $Z$ do not influence the corresponding value of $f_i^\theta$. Furthermore, $\tilde{s}_j$ coincides with $s_j$ for all $j \notin Z$, that is, for all components that might influence the value of $f_i^\theta(\tilde{s})$. More specifically, since we know that $f_i^\theta(s) = \theta$, we find $x, y \in [s]$ such that $K_{i,R_i(x)} = K_{i,min(s)} = 0$ and $K_{i,R_i(x)} = K_{i,max(s)} = 1$. If $z \in \{x, y\}$ does not lie in $[\tilde{s}]$, we can derive a sequence $z = z^1, \ldots, z^k$ in $[s]$ such that $z^k \in [\tilde{s}]$ and every $z^l$ differs from $z^{l-1}$ in one component $i_l \in Z$ only. Since there are no edges from components in $Z$ to $\alpha_i$ in $\mathcal{I}(s)$ we get $K_{i,R_i(z^l)} = K_{i,R_i(z)}$ for all $l \in \{1, \ldots, k\}$ according to Def. 3. Thus, we get that $K_{i,min(s)} = K_{i,min(\tilde{s})} = 0$ and $K_{i,max(s)} = K_{i,max(\tilde{s})} = 1$, and therefore $f_i^\theta(\tilde{s}) = f_i^\theta(s) = s_i = \tilde{s}_i$. □

The above lemma shows that the network behavior in $[s]$ is completely governed by the components of $\mathcal{I}^\theta(s)$. To give a clear understanding of how to construct the network dynamics from the dynamics derived from the subnetworks we need the following notation.

Let $s$ be a singular steady state and $Z$ a component of $\mathcal{I}^\theta(s)$ with $k := \operatorname{card} V_Z$. We may assume that $V_Z = \{\alpha_{l+1}, \ldots, \alpha_{l+k}\}$ for some $l \in \{0, \ldots, n-1\}$. Then $Z$ is an interaction graph

comprising $k$ vertices. Now, we want to define the dynamics of $Z$ as the projection of the dynamics of $\mathcal{I}$ with respect to $s$. We define a parameter set $K(Z)$ according to Def. 2.2 as the set of all parameters $K^Z_{i,R^Z_i(z)} := K_{i,R_i(\tilde{s})}$ for $z \in \mathcal{B}^k$ and $\tilde{s} \in \mathcal{B}^n$ with $\tilde{s}_i = s_i$ for all $i \notin J(s)$ and $\tilde{s}_i = z_{i-l}$ for all $i \in Z$. The parameters are well defined since there are no predecessors of vertices in $Z$ in $J(s) \setminus Z$. We set

$$f^{K(Z)} = f^Z : \mathcal{B}^k \to \mathcal{B}^k,\ z \mapsto (K^Z_{1,R^Z_1(z)}, \dots, K^Z_{k,R^Z_k(z)}).$$

We then have $f^Z = \pi^Z \circ f^\theta \circ \rho^Z$, where $\rho^Z : \mathcal{B}^k \to \mathcal{B}^n$ with $\rho^Z_i(z) = s_i$ for $i \notin Z$ and $\rho^Z_i(z) = z_{i-l}$ for $i \in Z$, and $\pi^Z : \mathcal{B}^n \to \mathcal{B}^k$ is the projection on the components of $Z$. Note that $f^Z$ yields always regular values, since the singular values in $J(s) \setminus Z$ do not influence the components in $Z$ according to Lem. 5.2. The definitions of parameters and $\mathcal{I}^\theta(s)$ ensure that all edges in $Z$ are functional. In a next step we can then derive the state transition graph $\mathcal{S}_Z = \mathcal{S}_{N^Z}$ for the network $N_Z = (Z, f^Z)$, the vertex set of which is $\mathcal{B}^k$.

We illustrate the definitions by considering our running example in Fig. 1. As shown in Fig. 3 (a), the graph $\mathcal{I}^\theta((0,\theta,\theta))$ has only one component $Z$ consisting of a positive circuit containing $\alpha_2$ and $\alpha_3$. We derive the parameters $K(Z)$ from those given in Fig. 1 for the global interaction graph. Since $s_1 = 0$, we obtain, according to the above definition, the parameters $K^Z_{2,\emptyset} := K_{2,\{e^-_{12},e^-_{32}\}} = 0$ and $K^Z_{2,\{e^+_{32}\}} := K_{2,\{e^-_{12},e^+_{32}\}} = 1$. The parameters for $\alpha_3$ stay the same, i. e., $K^Z_3, \omega = K_{3,\omega}$ for $\omega \in \{\emptyset, \{e^+_{23}\}\}$. The resulting state transition graph $\mathcal{S}^Z_N$ is also given in Fig. 3 (a).

After calculating the state transition graph for every component of $\mathcal{I}^\theta(s)$, we need to find a way to glue them together such that the resulting graph reflects the behavior of our original system in $[s]$.

**Definition 5.3.** *Let $s = (s_1, \dots, s_n)$ be a singular steady state and let $Z_1, \dots, Z_m$ be the components of $\mathcal{I}^\theta(s)$. W. l. o. g. we may assume that $Z_1$ contains the vertices $\alpha_1, \dots, \alpha_{\operatorname{card} Z_1}$, $Z_2$ contains the vertices $\alpha_{\operatorname{card} Z_1+1}, \dots, \alpha_{\operatorname{card} Z_1+\operatorname{card} Z_2}$, etc., and $s_k, \dots, s_n$ are all the regular components of $s$ for some $k \in \{1, \dots, n\}$. We then denote by $\mathcal{S}_{(s,Z_1,\dots,Z_m)}$ the graph with vertex set $V_{(s,Z_1,\dots,Z_m)} := V_{Z_1} \times \dots \times V_{Z_m} \times \{(s_k, \dots, s_n)\}$ and edge set $E_{(s,Z_1,\dots,Z_m)}$. An edge $s^1 \to s^2$ belongs to the edge set iff*

$$\pi^{Z_j}(s^1) = f^{Z_j}(\pi^{Z_j}(s^1)) = \pi^{Z_j}(s^2) \text{ for all } j \in \{1, \dots, m\},$$

*or if there exists $j \in \{1, \dots, m\}$ such that*

$$\pi^{Z_j}(s^1) \to \pi^{Z_j}(s^2) \text{ is an edge in } \mathcal{S}_{Z_j} \text{ and } s^1_i = s^2_i \text{ for all } i \notin V_{Z_j}.$$

*We call $\mathcal{S}_{(s,Z_1,\dots,Z_m)}$ the product state transition graph corresponding to $s$.*
*Furthermore, we denote by $\mathcal{S}^{[s]}_N$ the graph with vertex set $[s]$ and edge set $E_{[s]}$. For states $s^1$ and $s^2$ the edge $s^1 \to s^2$ is in $E_{[s]}$ iff it is an edge in the state transition graph $\mathcal{S}_N$ of $N$.*

Note that technically the sets $[s]$ and $V_{(s,Z_1,\dots,Z_m)}$ are not the same, but of course we can identify them with each other and do so to simplify notation. Since $s$ is a singular steady state, we can deduce from Lemma 5.2 that for each $x \in [s]$ we have $x' \in [s]$ for all $x'$ with $x'_i = f_i(x) \neq x_i$ for some $i$ and $x'_j = x_j$ for all $j \neq i$. Thus, there are no edges leaving $[s]$ in the state transition graph $\mathcal{S}_N$. The next theorem now tells us that we can reconstruct the behavior of the system $N$ in $[s]$ from the dynamics of the subnetworks $N_{Z_i}$.

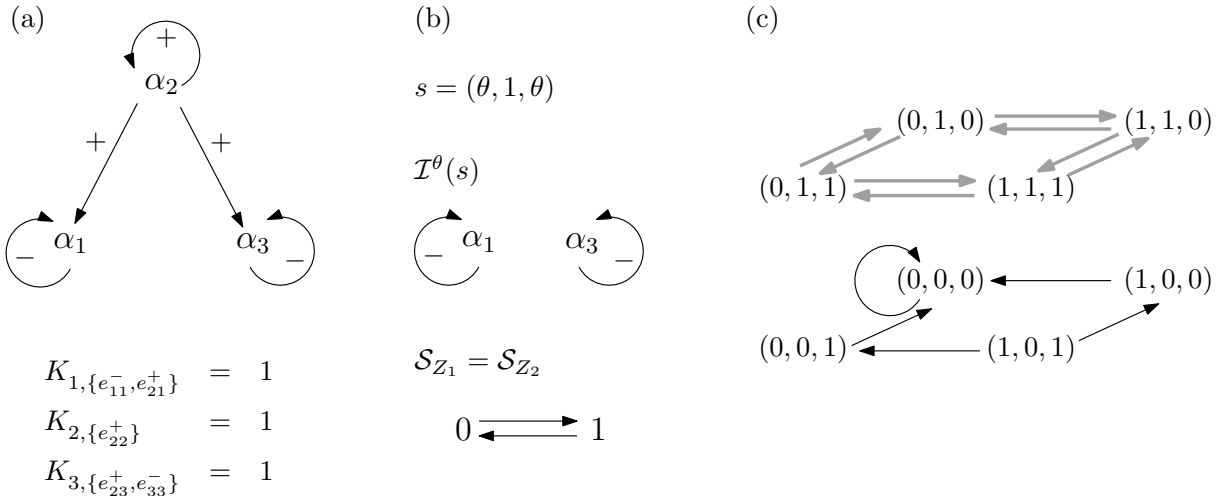



Figure 4: We list only the parameter values greater than zero in (a). The local interaction graph of $s$ in (b) has two components $Z_1$ and $Z_2$. In (c) the state transition graph of the system in (a).

**Theorem 5.4.** *Let $s = (s_1, \ldots, s_n)$ be a singular steady state and let $Z_1, \ldots, Z_m$ be the components of $\mathcal{I}^\theta(s)$. Then $\mathcal{S}_N^{[s]} = \mathcal{S}_{(s,Z_1,\ldots,Z_m)}$.*

*Proof.* As mentioned above, we identify the sets $[s]$ and $V_{(s,Z_1,\ldots,Z_m)}$. Let us remark that we have $\pi^{Z_j}(f(s')) = \pi^{Z_j}(f^\theta(s')) = \pi^{Z_j}(f^\theta(\rho^{Z_j}(\pi^{Z_j}(s')))) = f^{Z_j}(\pi^{Z_j}(s'))$ for all $s' \in [s]$ and all $j \in \{1, \ldots, m\}$ according to the definition of $\rho^{Z_j}$, $\pi^{Z_j}$ and Lemma 5.2.

Let $s^1 = (s_1^1, \ldots, s_n^1) \to s^2 = (s_1^2, \ldots, s_n^2)$ be an edge in $\mathcal{S}_N^{[s]}$. According to Def. 2.3, we have either $s^1 = s^2$ is a fixed point of $f$, or $s^1$ and $s^2$ differ in one component only. If $s^1 = f(s^1) = s^2$, we have $\pi^{Z_j}(s^1) = \pi^{Z_j}(s^2)$ and $f^{Z_j}(\pi^{Z_j}(s^1)) = \pi^{Z_j}(f(s^1)) = \pi^{Z_j}(s^1)$ for all $j \in \{1, \ldots, m\}$. Thus, $s^1 \to s^2$ is an edge in $\mathcal{S}_{(s,Z_1,\ldots,Z_m)}$.

Now, let us assume there exists $i \in \{1, \ldots, n\}$ such that $s_i^1 \neq f_i(s^1) = s_i^2$ and $s_j^1 = s_j^2$ for all $j \neq i$. Then there is $l \in \{1, \ldots, m\}$ such that $i \in Z_l$, and $s_j^1 = s_j^2$ for all $j \notin Z_l$. We have to show that $\pi^{Z_l}(s^1) \to \pi^{Z_l}(s^2)$ is an edge in $\mathcal{S}_{Z_l}$. Let $k^l$ be the cardinality of $V_{Z_l}$, and let $i_l \in \{1, \ldots, k^l\}$ such that $x_i = (\pi^{Z_l}(x))_{i^l}$ for all $x \in \mathcal{B}^n$, i. e. the $\pi^{Z_l}$ projects the $i$-th component of a $n$-vector on the $i^l$-th component of a $k$-vector. Then $f_{i^l}^{Z_l}(\pi^{Z_l}(s^1)) = (\pi^{Z_l}(f(s^1)))_{i^l} = f_i(s^1) = s_i^2 = (\pi^{Z_l}(s^2))_{i^l}$ and $f_{i^l}^{Z_l}(\pi^{Z_l}(s^1)) = f_i(s^1) \neq s_i^1 = (\pi^{Z_l}(s^1))_{i^l}$. Since we know $s_j^1 = s_j^2$ for all $j \neq i$, $j \in \{1, \ldots, n\}$, we have $(\pi^{Z_l}(s^1))_{j'} = (\pi^{Z_l}(s^2))_{j'}$ for all $j' \neq i^l$, $j' \in Z_l$. By definition $\pi^{Z_l}(s^1) \to \pi^{Z_l}(s^2)$ is an edge in $\mathcal{S}_{Z_l}$ and $s^1 \to s^2$ is an edge in $\mathcal{S}_{(s,Z_1,\ldots,Z_m)}$.

Similar straight forward reasoning shows that an edge in $\mathcal{S}_{(s,Z_1,\ldots,Z_m)}$ is also an edge in $\mathcal{S}_N^{[s]}$. □

For the system given in Fig. 4 the state $s = (\theta, 0, \theta)$ is a singular steady state. The components $Z_1$ and $Z_2$ of $\mathcal{I}^\theta(s)$ are the negative loop in $\alpha_1$ and the negative loop in $\alpha_3$, respectively. The state transition graphs $\mathcal{S}_{Z_1}$ and $\mathcal{S}_{Z_2}$ coincide and just consist of the cycle $0 \to 1 \to 0$. The graph $\mathcal{S}_{(s,Z_1,Z_2)}$ is indicated by heavier gray edges in the state transition graph in Fig. 4 (c). Looking at the example in Fig. 1 and the local interaction graphs and corresponding component

state transition graphs derived from the singular steady states $s^1 = (0, \theta, \theta)$ and $s^2 = (1, \theta, \theta)$ in Fig. 3, we see that the two product state transition graphs derived from them each form a component of the state transition graph of the original system. That is, we obtained the whole state transition graph from the subnetworks induced by the singular steady states. The following proposition generalizes this observation. The easy proof is omitted.

**Proposition 5.5.** *If $s^1, \ldots, s^k$ are singular steady states such that $\mathcal{B}^n = \bigcup_{i=1}^k [s^i]$, then $\mathcal{S}_N = \bigcup_{i=1}^k \mathcal{S}_N^{[s^i]}$.*

In many cases, the above proposition is not very useful. For the system in Fig. 3, for example, there is no singular steady state generating the lower component of the state transition graph. However, we are often not interested in transient states of the system, but rather in the asymptotic behavior. So, we do not necessarily want to reconstruct the complete state transition graph but rather focus on the attractors. According to Prop. 2.5 the graph $\mathcal{S}_{N^z}$ for a component of $\mathcal{I}^\theta(s)$ for a singular steady state $s$ contains an attractor. However, it is not a priori clear that attractors of subsystems generate attractors of the complete network. The next theorem shows that this is indeed possible. More precisely, all the attractors in $\mathcal{S}_N^{[s]}$ can be constructed from attractors of the subnetworks $Z_i$ and vice versa.

**Theorem 5.6.** *Let $s = (s_1, \ldots, s_n)$ and $Z_1, \ldots, Z_m$ be as in Def. 5.3. The attractors of $\mathcal{S}_N$ with vertices in $[s]$ are precisely the sets of states that can be represented as $A_1 \times \cdots \times A_m \times \{(s_k, \ldots, s_n)\}$, where for all $i \in \{1, \ldots, m\}$ the set $A_i$ denotes an attractor in $\mathcal{S}_{Z_i}$.*

*Proof.* For $i \in \{1, \ldots, m\}$ let $A_i$ be an attractor in $\mathcal{S}_{Z_i}$ and set $A := A_1 \times \cdots \times A_m \times \{(s_k, \ldots, s_n)\}$. We again interpret $A$ as a subset of $\mathcal{B}^n$. First, we show that $A$ is a trap set, i. e., every successor of a state in $A$ is again in $A$. Let $x \in A$ and $x'$ be a successor of $x$ in $\mathcal{S}_N$, and more precisely in $\mathcal{S}_N^{[s]}$. Assume $x \neq x'$. Since $\mathcal{S}_N^{[s]} = \mathcal{S}_{(s,Z_1,\ldots,Z_m)}$, we infer from Def. 5.3 that there exists $j \in \{1, \ldots, m\}$ such that $\pi^{Z_j}(x) \to \pi^{Z_j}(x')$ is an edge in $\mathcal{S}_{Z_j}$. Since $\pi^{Z_j}(x) \in A_j$ and since $A_j$ is an attractor, we have $\pi^{Z_j}(x') \in A_j$. Since $x$ and $x'$ only differ in one component, we have $x' \in A$.

Now, we have to show that there is a path from $x$ to $x'$ in $\mathcal{S}_N$ for all distinct $x, x' \in A$. First, we note that if there is an edge from state $z$ to state $z'$, $z \neq z'$, in $\mathcal{S}_{Z_l}$, $l \in \{1, \ldots, m\}$, then there is an edge from $x$ to $x'$ in $\mathcal{S}_N$ for all states $x, x' \in A$ satisfying $\pi^{Z_l}(x) = z$, $\pi^{Z_l}(x') = z'$, and $x_j = x'_j$ for all $j \notin Z_l$, according to the definition of edges in $\mathcal{S}_{(s,Z_1,\ldots,Z_m)} = \mathcal{S}_N^{[s]}$.

Let $x, x' \in A$. We set $x_i^1 := x_i$ for all $i \notin Z_1$ and $x_i^1 := x'_i$ for all $i \in Z_1$. For $l \in \{2, \ldots, m\}$ we set $x_i^l := x_i^{l-1}$ for all $i \notin Z_l$ and $x_i^l := x'_i$ for all $i \in Z_l$. Then there exists a path in $\mathcal{S}_{N^{Z_1}}$ from $\pi^{Z_1}(x)$ to $\pi^{Z^1}(x^1)$, since $A_1$ is an attractor. As seen above, we then can find a path $\gamma_1$ from $x$ to $x^1$ in $\mathcal{S}_N$ such that $\tilde{x}_j = x_j$ for every state $\tilde{x} \in \gamma_1$ and every $j \notin Z_1$. In the same fashion we find a path $\gamma_2$ from $x^1$ to $x^2$ in $\mathcal{S}_n$ such that $\tilde{x}_j = x_j^1$ for all $\tilde{x} \in \gamma_2$ and $j \notin Z_2$. We continue the procedure for $Z_3, \ldots, Z_m$. Since $x^m = x'$ per definition, combining the paths $\gamma_i$ in the order of their indices yields a path from $x$ to $x'$ in $\mathcal{S}_N$. It follows that $A$ is an attractor in $\mathcal{S}_N$.

Now, let $A$ be in arbitrary attractor in $\mathcal{S}_N$ with vertices in $[s]$ and consider the set $A_j = \pi^{Z_j}(A)$ for $j \in \{1, \ldots, m\}$. $A_j$ has to be a trap set in $\mathcal{S}_{Z_j}$ since edges leaving $A_j$ would generate edges leaving $A$ in $\mathcal{S}_{(s,Z_1,\ldots,Z_m)} = \mathcal{S}_N^{[s]}$ according to Def. 5.3. Let $x, x' \in A_j$. Then there exist states $y, y' \in A$ such that $x = \pi^{Z_j}(y)$ and $x = \pi^{Z_j}(y)$. Since $A$ is an attractor, there is a path $\gamma$ from $y$ to $y'$ in $\mathcal{S}_N$, and more precisely in $\mathcal{S}_N^{[s]}$, since $[s]$ is a trap set. It is easy to verify that

we obtain a path from $x$ to $x'$ in $\mathcal{S}_{N_j}$ by projecting the vertices of $\gamma$ onto $Z_j$ and eliminating all but one of consecutive identical vectors in the resulting sequence of states. The existence of edges in $\mathcal{S}_{N_j}$ between the remaining vertices is again guaranteed by Def. 5.3. □

Let us use the example in Fig. 1 to illustrate the theorem. In Fig. 3 (a), we see the state transition graph $\mathcal{S}_N^Z$ for the single component $Z$ of $\mathcal{I}^\theta(s)$. It contains the attractors $\{(0,0)\}$ and $\{(1,1)\}$. It follows from Theorem 5.6 that the sets $\{(0,0,0)\}$ and $\{(0,1,1)\}$ are attractors in $\mathcal{S}_N$. Similarly, we derive a state transition graph from $\mathcal{I}^\theta((1,\theta,\theta))$ which consists of a negative circuit. The state transition graph is shown in Fig. 3 (b) and contains only one attractor, the set $\{(0,0),(1,0),(1,1),(0,1)\}$, which has cardinality greater than one. Thus, we find an attractor $\{(1,0,0),(1,1,0),(1,1,1),(1,0,1)\}$ in $\mathcal{S}_N$. The state transition graph $\mathcal{S}_N$ is given in Fig. 1 with the attractors emphasized.

The number and size of attractors are important characteristics of bioregulatory networks. There are many results that link these dynamical characteristics to properties of the network structure. In [3] it is shown that isolated circuits always display a characteristic behavior depending on their sign. A positive circuit gives rise to two attractors, more precisely two steady states, a negative circuit results in a cyclic attractor, i. e., an attractor with cardinality greater than one. The situation is much more difficult to analyze if there are many circuits in $\mathcal{I}$, possibly even intertwined. Thomas conjectured in 1981 that the existence of a positive resp. negative circuit in the interaction graph is a necessary condition for the existence of two attractors resp. a cyclic attractor in the state transition graph. The conjectures haven been proven in different settings (see e. g. [12], [4] and [6]). For regulatory networks without context sensitivity, we formulated in [10] a sufficient condition for circuits to display their characteristic behavior using singular steady states. The proof in [10] can be easily adapted to show the next statement.

**Lemma 5.7.** *Let $I$ be an interaction graph that contains only one circuit $C$. If $C$ is a positive circuit, then $f$ has two fixed points. If $C$ is negative, then there exists an attractor with cardinality greater than one in the state transition graph.*

We make some short remarks on the proof. Recall our assumption that every vertex in $\mathcal{I}$ has a predecessor. Since every edge is functional, the state $(\theta,\ldots,\theta)$ is steady. In [10], it is shown that $\mathcal{I}$ then has a particular structure. It consists of the circuit $C$ with directed acyclic graphs coming out of vertices of $C$. This structure allows us to explicitly specify values for the vertices of $C$ that remain fix under $f^\theta$ in the case of $C$ being positive, or behave like a trap cycle, if $C$ is negative. From this core behavior we can then infer the behavior of the whole graph. Here, we also have to consider that there may be parallel edges outside the circuit $C$. However, the proof method is still valid. The necessary technical adaptations to the proofs in [10] correspond to those made in the proof of Lemma 5.2.

The above lemma together with Theorem 5.6 leads to the following theorem. Recall that we denote the component of $\mathcal{I}^\theta(s)$ containing some circuit $C$ by $J_C(s)$.

**Theorem 5.8.** *Let $C$ be a circuit in $\mathcal{I}$ and $s$ a singular steady state characteristic of $C$. Assume that $C$ is the only circuit in the component $J_C(s)$ of $\mathcal{I}^\theta(s)$. If $C$ is a positive circuit, then $f^\theta$ has at least three fixed points and $\mathcal{S}_N$ contains at least two attractors. If $C$ is negative, there is an attractor in $\mathcal{S}_N$ with cardinality greater than one.*

*Proof.* We may assume that $J_C(s)$ comprises the vertices $\alpha_1,\ldots,\alpha_r$ for some $r \in \{1,\ldots,N\}$. Let at first $C$ be positive. Then $f^{J_C(s)}$ has two fixed points $x, x' \in \mathcal{B}^r$ according to Lemma 5.7.

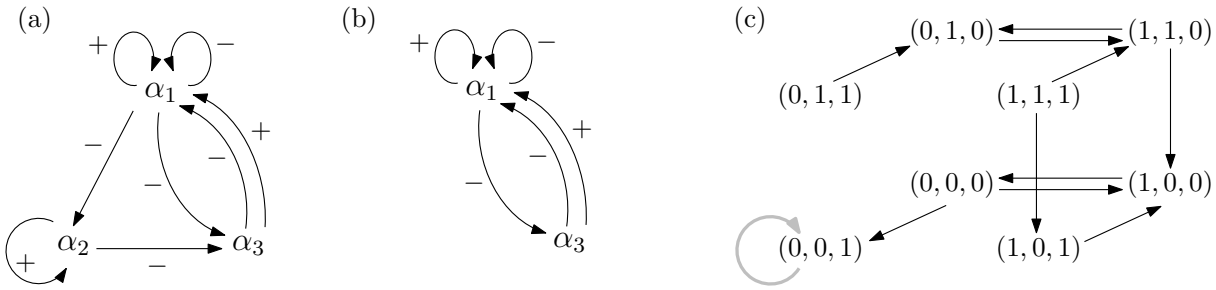



Figure 5: We choose the parameters for the interaction graph in (a) as $K_{1,\{e_{11}^+,e_{31}^+\}} = K_{1,\{e_{11}^-,e_{31}^-\}} = K_{2,\{e_{12}^-,e_{22}^+\}} = K_{3,\{e_{13}^-,e_{23}^-\}} = 1$ and set all other parameters 0. In (b) the graph $\mathcal{I}^\theta(s)$ for the singular steady state $s = (\theta, 0, \theta)$. In (c) the corresponding state transition graph.

We define states $s^1$ and $s^2$ in $\mathcal{B}_\theta^n$ by $s_i^1 := s_i^2 := s_i$ for all $i \notin J_C(s)$, $s_i^1 := x_i$ and $s_i^2 := x_i'$ for all $i \in \{1, \ldots, r\}$. From Lemma 5.2 follows that the states $s^1$ and $s^2$ are steady states. Thus $f^\theta$ has three fixed points, since $s$ is distinct from $s^1$ and $s^2$. According to Theorem 5.6 we find attractors $A_1$ and $A_2$ in $\mathcal{S}_N$ such that $\pi^{J_C(s)}(A_1) = \{s^1\}$ and $\pi^{J_C(s)}(A_2) = \{s^2\}$.

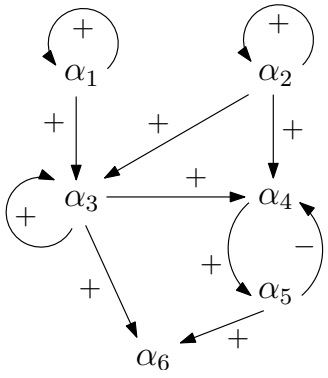
If $C$ is negative, we find an attractor $A'$ in the state transition graph of the component graph $J_C(s)$ with $\operatorname{card} A' > 1$. Theorem 5.6 yields an attractor $A$ in $\mathcal{S}_N$ with $\pi^{J_C(s)}(A) = A'$. Thus cardinality of $A$ is also greater than one. □

Theorem 5.8 is a stronger result than the one obtained in [10], even for networks without context sensitivity. The use of local interaction graphs allows for a more refined picture of the dynamics possible in restricted parts of the state space.

Our running example from Fig. 1 together with Fig. 3 illustrates the theorem. Figure 5 shows that the statement does not hold, if the circuit $C$ is not the only circuit in $J_C(s)$. The state $(\theta, 0, \theta)$ is steady for the bioregulatory network derived from the interaction graph in (a) and the parameters specified in the caption. There are four circuits in $\mathcal{I}^\theta((\theta, 0, \theta))$, two negative and two positive circuits. However, the state transition graph contains only one attractor, namely the set $\{(0, 0, 1)\}$, as is shown in (c). Neither the behavior characteristic for positive circuits nor that characteristic for negative circuits is displayed. Further examples can be found in [10]. However, a system may display the behavior characteristic for a circuit of a given sign, although there is no singular steady $s$ such that the circuit is the only one in the corresponding component of $\mathcal{I}^\theta(s)$. The condition is not necessary, as illustrated by an example given in [10], Fig. 4.

## 6 Networks with Input Layer

In the preceding section we gave several results on how to derive information on the behavior of a complex network by looking at suitable subnetworks, which in turn we derived from singular steady states. However, in most cases the results do not yield a complete analysis of the system's dynamics. To obtain more comprehensive results we are faced with two difficulties. First, we need to make sure that our in nature local analysis covers all parts of state space containing asymptotically stable behavior. This is a difficult task, since we in general do not know anything about the network dynamics a priori. This directly relates to the second difficulty. Since we use singular steady states to deconstruct the network, we need a method to find singular steady




| $(s_1, s_2)$ | $f(s) = s$ | attractors |
|---|---|---|
| $(0,0)$ | $(0,0,\theta,0,0,\theta)$ | $\{(0,0,0,0,0,0)\}, \{(0,0,1,0,0,1)\}$ |
| $(1,0)$ | $(1,0,1,0,0,1)$ | $\{(1,0,1,0,0,1)\}$ |
| $(1,1)$ | $(1,1,1,\theta,\theta,1)$ | $\{(1,1,1,x,y,1) \mid x,y \in \{0,1\}\}$ |
| $(0,1)$ | $(0,1,1,\theta,\theta,1)$ | $\{(0,1,1,x,y,1) \mid x,y \in \{0,1\}\}$ |

Figure 6: We choose the parameters for the given interaction graph such that the inputs to $\alpha_3$ and $\alpha_6$ operate via an OR gate, while inputs to $\alpha_4$ are processed via an AND gate. The left column in the table lists the different input values for the network, the middle column the derived singular steady states, and the right column the resulting attractors.

states, and, coming back to the first problem, in particular to identify a set of singular steady states such that the generated subnetworks hold sufficient information to characterize the global dynamics of the whole system.

In this section we introduce a class of networks for which we can solve the problems described above. In the following, we only consider interaction graphs the underlying undirected graph of which is connected. For the general case, we just consider the components separately.

**Definition 6.1.** *We call $N$ a* network with input layer*, if there exists a vertex $\alpha_i$ with a positive edge to itself and no other incoming edges. A vertex satisfying this condition is called* input vertex.

In the following we assume that $N$ is a network with input layer. Without loss of generality we assume that $\alpha_1, \ldots, \alpha_k$ are the input vertices of $N$. In Fig. 6 we see a network with input layer. The components $\alpha_1$ and $\alpha_2$ are the input vertices.

Networks with input layer often play an important role in biological systems. When modeling signal transduction networks, for example, we can model receptors as input vertices. Different input values then represent different signals reaching the receptors, and we want to understand how the system reacts to such signals.

If a system has an input layer, then we can immediately make some observation about the network dynamics. The coordinate function $f_i$ governing the behavior of the input vertex $\alpha_i$ solely depends on the value of $\alpha_i$. Since $\alpha_i$ influences itself via a positive edge, we have $f_i(x) = x_i$ for all $x \in \mathcal{B}^n$ according to condition (1). That is, the input vertex values always stay fixed. Thus the state transition graph consists of at least $2^k$ components, and in every component the state component values corresponding to input vertices stay fixed. In particular, we know that each attractor of the system is contained in one of the components. This proves the statement of the following lemma.

**Lemma 6.2.** *Let $A$ be an attractor in $\mathcal{S}_N$. Then $a_i = a'_i$ for all $a, a' \in A$ and $i \in \{1, \ldots, k\}$.*

The specific structure of a network with input layer allows us to find singular, or possibly even regular, steady states in a simple way. Recall that we denote the set of singular components of a state $s$ by $J(s)$.

**Lemma 6.3.** *Let $x_1, \dots, x_k \in \mathcal{B}$. Define $x^0 \in \mathcal{B}_\theta^n$ by $x_i^0 := x_i$ for all $i \in \{1, \dots, k\}$ and $x_i^0 := \theta$ for $i > k$. Then the sequence $(x^l)_{l\in\mathbb{N}}$ where $x^l := f^\theta(x^{l-1})$ converges to a fixed point $s$ of $f^\theta$. Moreover, if we set $x_i^0 := x_i$ for all $i \in \{1, \dots, k\}$ and arbitrarily choose $x_i^0 \in \mathcal{B}$ for $i > k$ and define the sequence $(x^l)_{l\in\mathbb{N}}$ as above, then there exists $l \in \mathbb{N}$ such that $x_i^m = s_i$ for all $i \in \{1, \dots, n\} \setminus J(s)$ and $m \geq l$.*

*Proof.* First, we show by induction that if $x_i^l$ is a regular value for some $l \in \mathbb{N}$ and some $i \in \{1, \dots, n\}$, then $x_i^m = x_i^l$ for all $m \geq l$. The only regular values of $x^0$ correspond to the input vertices $\alpha_1, \dots, \alpha_k$, and we have $x_i^m = x_i^0$ for all $m \in \mathbb{N}$ and $i \in \{1, \dots, k\}$. Let $l \in \mathbb{N}$. We assume that if $x_i^l \in \{0, 1\}$ for some $i \in \{1, \dots, n\}$, then $x_i^m = x_i^l$ for all $m \geq l$.

Let $i \in \{1, \dots, n\}$ such that $x_i^{l+1} \in \{0, 1\}$. If $i \notin J(x^l)$, our assumption yields $x_i^m = x_i^l = x_i^{l+1}$ for all $m \geq l + 1$. Let $i \in J(x^l)$. Since $x_i^{l+1} = f_i^\theta(x^l)$ is a regular value, we have $f_i^\theta(x^l) = K_{i,\min(x^l)} = K_{i,\max(x^l)}$. Since $x^l$ and $x^m$ coincide in all regular values of $x^l$ for all $m \geq l$, we have $[x^m] \subseteq [x^l]$ for all $m \geq l$. It follows from the definition of $K_{i,\min(x^m)}$ and $K_{i,\max(x^m)}$ that $K_{i,\min(x^l)} \leq K_{i,\min(x^m)} \leq K_{i,\max(x^m)} \leq K_{i,\max(x^l)}$, and thus $K_{i,\min(x^m)} = K_{i,\max(x^m)}$ for all $m \geq l$. We obtain $x_i^{m+1} = f_i^\theta(x^m) = K_{i,\min(x^m)} = K_{i,\min(x^l)} = x_i^{l+1}$ for all $m \geq l$, which proves our statement. Since we obtain the sequence $(x^l)_{l\in\mathbb{N}}$ by iteration of $f^\theta$ and since $\mathcal{B}_\theta^n$ is finite, the sequence converges to a regular or singular fixed point of $f^\theta$.

Now set $\tilde{x}_i^0 := x_i^0$ for $i \in \{1, \dots, k\}$, choose $\tilde{x}_i \in \mathcal{B}$ for $i > k$ and define $\tilde{x}^{l+1} := f(\tilde{x}^l)$ for $l \in \mathbb{N}$. Then, we have $\tilde{x}^0 \in [x^0]$ and thus it follows with the same reasoning as above that $\tilde{x}_i^1 = x_i^1$ for all $i \notin J(x^1)$. Again, we have $\tilde{x}^1 \in [x^1]$ and it follows $\tilde{x}_i^2 = x_i^2$ for all $i \notin J(x^2)$. Repeating the argument until we reach the fixed point $s$ generated by $(x^l)_{l\in\mathbb{N}}$, we find $l \in \mathbb{N}$ such that $\tilde{x}_i^m = s_i$ for all $m \geq l$ and $i \notin J(s)$. □

We call the resulting fixed point in the preceding lemma the fixed point *derived from the input values* $x_1, \dots, x_k$. It is easy to see that for such a fixed point $s$ the set $[s]$ is a trap set, and thus contains at least one attractor.

The above lemma provides a method to translate combinations of input values into singular or regular steady states. It turns out that if we calculate the fixed points derived from all possible combinations of input values, then we can completely describe the asymptotic behavior of the network by considering resulting regular steady states and the subnetworks obtained from the resulting singular steady states.

**Theorem 6.4.** *Let $A$ be an attractor of $N$. Then there exist input values $x_1, \dots, x_k \in \mathcal{B}$ such that either $A = \{s\}$ or we can construct $A$ from $s$ as shown in Theorem 5.6, where $s = (s_1, \dots, s_n)$ is the fixed point derived from $x_1, \dots, x_k$.*

*Proof.* Let $a' \in A$ and set $x_i := a'_i$ for all $i \in \{1, \dots, k\}$. Let $s$ be the fixed point derived from $x_1, \dots, x_k$. According to Lemma 6.2, the first $k$ components remain fix in $A$, and thus $a_i = x_i = s_i$ for all $i \in \{1, \dots, k\}$. Since $A$ is a trap set, we can deduce from the second statement in Lemma 6.3 that $a_i = s_i$ for all $a \in A$ and $i \in \{1, \dots, n\} \setminus J(s)$, since those values remain fixed under $f$ after sufficiently many iteration steps. It follows that $A \subseteq [s]$. If $s$ is a singular steady state, then we can construct $A$ from the attractors of the subnetworks derived from $s$ according to Theorem 5.6. Otherwise, we have $J(s) = \emptyset$ and thus $A$ consists of the regular fixed point $S$. □

We illustrate the theorem with the system given in Fig. 6. The table shows in the left column the possible combinations of input values, in the middle column the fixed point derived

from those values, and in the right column the resulting attractors of the system. The input $(\alpha_1, \alpha_2) = (0, 0)$ yields the singular steady state $(0, 0, \theta, 0, 0, \theta)$. The subnetwork governing the behavior of the system in the corresponding part of state space consists of the positive loop in $\alpha_3$ and the edge from $\alpha_3$ to $\alpha_6$. This subsystem has two fixed points $(0, 0)$ and $(1, 1)$. Thus, we obtain two steady states of the original system, namely $(0, 0, 0, 0, 0, 0)$ and $(0, 0, 1, 0, 0, 1)$. The input $(1, 0)$ directly leads to the regular steady state $(1, 0, 1, 0, 0, 1)$. Both input vectors $(1, 1)$ and $(0, 1)$ generate a singular steady state whose local interaction graph consists of the negative circuit between $\alpha_4$ and $\alpha_5$. This subsystem's only attractor is $\{(0, 0), (0, 1), (1, 1), (1, 0)\}$. We obtain a cyclic attractor of the original system for each of the two input vectors. According to Theorem 6.4 the five listed attractors are all attractors of the system.

# 7 Conclusion

The focus of this paper is to obtain information about the dynamics of complex regulatory networks by analyzing subsystems of the network, thereby reducing the complexity of the problem. To identify suitable subnetworks we use the notion of singular steady state, which we introduced in [10] in a more restricted framework. In addition we employ the idea of local interaction graphs in order to obtain a refined understanding of what interactions in the network structure measurably influence the behavior of the system in any given part of state space. We show in Sect. 5 that the information about the network behavior encoded in a given singular steady state together with the refined structural representation of a corresponding interaction graph allows us to derive the dynamics of the whole network, at least in part of state space, and show how to construct attractors of the network from attractors of subnetworks. This approach also yields a deeper understanding of the relation between structural and dynamical network characteristics. We obtain a result linking the existence of circuits in the interaction graph to the existence of multiple attractors resp. an attractor with cardinality greater than one, which generalizes and refines a corresponding statement in [10]. Feedback circuits are known to be an important building block, or so-called network motif, for bioregulatory networks. Our result gives sufficient conditions for them to imprint the behavior characteristic for them in isolation on a complex network containing such circuits. Other important structural network motifs have been characterized, and in this context application of our methods may help to analyze their behavior if embedded in a complex network.

In future work, we also plan to focus more on the application side. Sect. 6 introduces a solid starting point for this endeavor. In particular, the analysis of signal transduction networks is of interest for further studies. However, when modeling biological systems, Boolean networks, although often yielding a fruitful first model, can only provide a very course description. Multi-valued models allow for finer representation, still preserving the advantages of discrete modeling. Generalizing the results presented in this paper to the framework of multi-valued discrete networks thus is another priority for future work.

# References

[1] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. Application of formal methods to biological regulatory networks: extending Thomas' asynchronous logical approach with temporal logic. *J. Theoret. Biol.*, 229:339–347, 2004.

[2] C. Chaouiya, E. Remy, B. Mossé, and D. Thieffry. Qualitative analysis of regulatory graphs: a computational tool based on a discrete formal framework. In *First Multidisciplinary International Symposium on Positive Systems: Theory and Applications, POSTA 2003*, volume 294 of *LNCIS*, pages 119–126. Springer, 2003.

[3] E. Remy, B. Mossé, C. Chaouiya, and D. Thieffry. A description of dynamical graphs associated to elementary regulatory circuits. *Bioinformatics*, 19:172–178, 2003.

[4] E. Remy and P. Ruet. On differentiation and homeostatic behaviours of Boolean dynamical systems. In *Transactions on Computational Systems Biology VIII*, volume 4780 of *LNCS*, pages 92–101. Springer, 2007.

[5] E. Remy, P. Ruet, and D. Thieffry. Graphic requirements for multistability and attractive cycles in a boolean dynamical framework. *Adv. Appl. Math.*, 41(3):335–350, 2008.

[6] A. Richard and J.-P. Comet. Necessary conditions for multistationarity in discrete dynamical systems. *Discr. Appl. Math.*, 155(18):2403–2413, 2007.

[7] A. Richard, J.-P. Comet, and G.Bernot. R. Thomas' modeling of biological regulatory networks: introduction of singular states in the qualitative dynamics. *Fund. Inform.*, 65:373–392, 2005.

[8] F. Robert. *Discrete Iterations: A Metric Study*, volume 6 of *Springer Series in Computational Mathematics*. Springer, 1986.

[9] H. Siebert. Local structure and behavior of boolean bioregulatory networks. In *Algebraic Biology, AB 2008, Castle of Hagenberg, Austria*, volume 5147 of *LNCS*, pages 185–199. Springer, 2008.

[10] H. Siebert and A. Bockmayr. Relating attractors and singular steady states in the logical analysis of bioregulatory networks. In *Algebraic Biology, AB 2007, Castle of Hagenberg, Austria*, volume 4545 of *LNCS*, pages 36–50. Springer, 2007.

[11] E. H. Snoussi and R. Thomas. Logical identification of all steady states: the concept of feedback loop characteristic states. *Bull. Math. Biol.*, 55:973–991, 1993.

[12] C. Soulé. Graphical requirements for multistationarity. *ComPlexUs*, 1:123–133, 2003.

[13] R. Thomas and R. d'Ari. *Biological Feedback*. CRC Press, 1990.

[14] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. II. Logical analysis of regulatory networks in terms of feedback circuits. *Chaos*, 11:180–195, 2001.

# Chapter 4

# Analysis of Discrete Regulatory Networks Using Symbolic Steady States

Heike Siebert.

*Bulletin of Mathematical Biology*, 73 (4): 873-898, 2011.

**Remarks.** Interpreting singular states as a symbolic representation of a set of states allows for transference of the ideas in the previous chapters to multi-valued update functions representing regulatory networks. I am not only able to generalize many of the results in the Boolean setting to this richer framework, but I also broaden the scope by including results for synchronous as well as asynchronous dynamics. Moreover, I provide an in-depth analysis of the conditions needed to generate symbolic steady states, linking structural to dynamical properties.

# Analysis of Discrete Bioregulatory Networks Using Symbolic Steady States

Heike Siebert

*Abstract.* A discrete model of a biological regulatory network can be represented by a discrete function that contains all available information on interactions between network components and the rules governing the evolution of the network in a finite state space. Since the state space size grows exponentially with the number of network components, analysis of large networks is a complex problem. In this paper, we introduce the notion of symbolic steady state that allows us to identify subnetworks that govern the dynamics of the original network in some region of state space. We state rules to explicitly construct attractors of the system from subnetwork attractors. Using the results, we formulate sufficient conditions for the existence of multiple attractors resp. a cyclic attractor based on the existence of positive resp. negative feedback circuits in the graph representing the structure of the system. In addition, we discuss approaches to finding symbolic steady states. We focus both on dynamics derived via synchronous as well as asynchronous update rules. Lastly, we illustrate the results by analyzing a model of T helper cell differentiation.

## 1 Introduction

Discrete methods of modeling biological regulatory networks are often used if the available data is rather qualitative in nature. Each component of the network is associated with a finite number of activity levels representing e. g. a concentration interval of a substance, activity of a gene or presence or absence of a signal. The state space of the system then consists of vectors of the component activity levels, and the network dynamics is derived from a discrete function $f$ capturing the rules of component interactions in the system. Here, two fundamentally different methods of calculating trajectories of the system are in use. The so-called synchronous update renders a deterministic representation by defining the successor of a given state as its image under $f$. In contrast, the asynchronous update method yields non-deterministic dynamics. Motivated by the assumption of distinct time delays associated with component value changes, we require that a state and its successor differ in one component only, but consider all successor possibilities in agreement with $f$. Both approaches have been used successfully, the synchronous method having advantages regarding the complexity of the analysis, the asynchronous update often allowing for a more realistic representation of the system's behavior (see e. g. [11], [35], [12] and references therein).

When analyzing large and complex networks, one is often interested in identifying subnet-

works that are in some sense significant for the system. The idea of network decomposition is well-established in systems biology, and has been approached from many different angles (see e. g. [9, 4, 20, 19, 1]). We are particularly interested in subsystems that play crucial roles in the dynamics of the system. Analysis of such networks in isolation may then yield information on the dynamical behavior of the original network. Clearly, the difficulty is that further components and interactions influence such a network building block once it is again embedded in the network. Conditions to identify suitable subnetworks that retain their behavior once re-embedded are needed to derive useful information on the network dynamics.

In this paper we generalize and extend corresponding ideas developed for Boolean functions and asynchronous dynamics in [32] to multi-valued discrete functions, considering synchronous as well as asynchronous dynamics. By exploiting the properties of *symbolic steady states*, we identify regions of state space where a number of network components remain fixed in the dynamics independent of the values of the remaining components. These are fixed points of an adapted function $f^\theta$ that coincides with $f$ for the most part, but also allows the consideration of a symbolic value $\theta$ for the network components. The value $\theta$ can be identified with the whole activity level range of a given component, representing uncertainty of the actual value of that component in a network state. The *regular* components, i. e. those with a specified activity level, of a symbolic steady state act as a boundary between dynamically active subnetworks similar to the notion of *frozen core* introduced in the context of random Boolean networks (see [11]). We obtain a detailed structural representation of the active subnetworks by considering the local interaction graph associated with the symbolic steady state. Assembly of attractors of the isolated subnetworks with respect to the symbolic steady state then yields attractors of the original network. With this fundamental property in mind, we are able to proof more general statements concerning the relation between network structure and network dynamics. Here, we proof that the existence of a positive resp. a negative circuit under certain conditions implies the existence of multiple attractors resp. a cyclic attractor in the synchronous as well as the asynchronous dynamics.

The results mentioned thus far concern analysis using symbolic steady states. We also address the problem of determining symbolic steady states, exploiting structural as well as dynamical characteristics of $f$. All results are illustrated using a model of T helper cell differentiation.

This paper is organized as follows. In the next section we introduce the modeling framework used in this paper. In Sect. 3 we establish the notion of symbolic steady state as well as some important properties, followed by results on compositional attractors of subnetworks derived from symbolic steady states in Sect. 4. The results are then used to obtain statements linking the existence of feedback circuits in the network structure to number and size of attractors in Sect.5. In Sect. 6, we examine different methods for deriving symbolic steady states. The results are applied to a model of T helper cell differentiation in Sect. 7. We end the paper with concluding remarks and perspectives for future work.

## 2 Regulatory Networks

Throughout the text let us consider a network with $n \in \mathbb{N}$ components $\alpha_1, \ldots, \alpha_n$. To simplify notation, we identify each component $\alpha_i$ with its index $i$. Each component is understood as a discrete variable whose values signify the different activity levels of the network component. Activity levels may represent different biological characteristics, e. g. substance concentration,

gene activity, absence or presence of a signal and so on. The number of activity levels of different components may differ, depending on function of components and available data. Thus, every component $\alpha_i$ is associated with a *range* $X_i := \{0, 1, \ldots, p_i\}$ of activity levels, where $p_i \in \mathbb{N}$ denotes the maximal activity level of $\alpha_i$. The set $X := X_1 \times \cdots \times X_n$ comprises all possible activity level vectors and thus represents the state space of the system. Interaction of network components and rules governing the network's dynamics are then captured by a discrete function $f = (f_1, \ldots, f_n) : X \to X$. If $p_i = 1$ for all maximal activity levels $p_i$, then $f$ is a Boolean function.

## 2.1 Structure

In a next step, we want to derive the network structure from the function $f$. As commonly done, we represent the structure as a signed directed (multi-)graph with vertex set $V := \{\alpha_1, \ldots, \alpha_n\}$ and edges representing interactions between components. The sign of an edge describes the character of the interaction, a negative sign signifying an inhibiting, a positive sign an activating effect. However, in some cases the influence of one component on another depends on the current state of the network. For example, if two substances form a complex that in turn activates some target gene, then in general the presence of only one of those substances is not sufficient to induce gene expression. So, one of the substances can only effectively influence the gene when the other substance is present. Another possibility is that the character of an interaction changes depending on the state of the network. A well-known example is the DNA-binding protein TCF which can be involved in repression as well as activation of the same target genes (see e. g. [31]). Such refined structural information is of great interest when linking structural and dynamical aspects and thus we want to include it in the structural representation of the network. This is done by considering *local interaction graphs*. This notion was introduced for Boolean functions in [24] and is used for multi-value functions in the form considered here in [26]. In the following we denote with sgn the usual sign function that takes values in $\{-1, 0, +1\}$. When talking about edge signs we identify the symbols $-1$ and $-$ as well as $+1$ and $+$.

**Definition 2.1.** *Let $x \in X$. By $G(x) := G(f)(x)$ we denote the directed signed (multi-)graph with vertex set $\{\alpha_1, \ldots, \alpha_n\}$ and edge set $E(x) \subseteq V \times V \times \{+, -\}$. An edge $(i, j, \varepsilon)$ belongs to $E(x)$ iff there exists $c_i \in \{-1, +1\}$ such that $x_i + c_i \in X_i$ and*

$$\operatorname{sgn} \frac{f_j((x_1, \ldots, x_{i-1}, x_i + c_i, x_{i+1}, \ldots, x_n)) - f_j(x)}{c_i} = \varepsilon \,.$$

*We call $G(x)$ the* local interaction graph of $f$ in $x$.

To obtain the local interaction graph, we consider changes in the values of the coordinate functions depending on small changes, i. e. changes by absolute value 1, in one component. The local interaction graph in $x$ is thus closely related to the discrete Jacobian matrix, which was introduced in [29] in the Boolean case. Note that in the multi-value other than in the Boolean case it is possible that $G(x)$ contains parallel edges. By definition, there are at most two parallel edges from one vertex to another which then have opposite sign, one resulting from an increase, the other from a decrease of the component value in Def. 2.1.

If we combine the structural information of the local interaction graphs for a set of states $M$ we obtain a graph that contains all interactions influencing the network's dynamics in $M$.

**Definition 2.2.** *Let $M \subseteq X$. We denote by $G(M) := G(f)(M)$ the union of the graphs $G(x)$, $x \in M$. For $M = X$ we set $G(f) := G(X)$ and call $G(f)$ the* global interaction graph of $f$.

The global interaction graph contains all interactions influencing the network's dynamics.

When analyzing interaction graphs, we are interested in certain structural motifs. Mainly, we focus on so-called *(feedback) circuits*. Here, a circuit is a tuple $(e_1, \ldots, e_r)$ of edges $e_i = (k^i, l^i, \varepsilon^i) \in E$ such that all $k^i$, $i \in \{1, \ldots, r\}$, are pairwise distinct, and $l^i = k^{i+1}$ for all $i \in \{1, \ldots, r\}$ modulo $r$. The *sign of a circuit* is the product of the signs of its edges. Note that in a multigraph a circuit is not uniquely determined by its vertices.

In Fig. 1 we see on the left interaction graphs of the function $f = (f_1, f_2, f_3) : X \to X$, $X := \{0,1\}^2 \times \{0,1,2\}$, introduced in the caption. Here, $f_2$ models that $\alpha_2$ is influenced by $\alpha_1$ and $\alpha_3$ via an OR-gate as long as the activity level of $\alpha_3$ is below 2. However, if $\alpha_3 = 2$, then $\alpha_2$ is repressed. The way $\alpha_3$ influences $\alpha_2$ thus depends on the current state of the system. If the system is in state $(1,0,0)$, then a small change in the $\alpha_3$ value is not enough to reach value 2. Thus, $f_2$ corresponds to a logical OR-function. Since $\alpha_1 = 1$, we have $f_2((1,0,x_3)) = 1$ for $x_3 \in \{0,1\}$. It follows that in state $(1,0,0)$ small changes in $\alpha_3$ do not influence the component value $\alpha_2$. As a result, there is no edge from $\alpha_3$ to $\alpha_2$ in $G((1,0,0))$ in Fig. 1 (a). In comparison, if we look at state $(0,0,0)$, we have $f_2((0,0,1)) - f_2((0,0,0)) = 1$ and therefore we get a positive edge from $\alpha_3$ to $\alpha_2$ in $G((0,0,0))$. Lastly, when looking at state (0,0,1), $\alpha_3$ influences $\alpha_2$ via a positive edge, since the above argument is still valid. However, if we increase the activity level of $\alpha_3$ to 2, then by definition of $f_2$ we have a negative influence of $\alpha_3$ on $\alpha_2$. Thus in the local interaction graph $G((0,0,1))$ there is a negative as well as a positive edge from $\alpha_3$ to $\alpha_1$. By definition, all the local interaction graphs are subgraphs of the global interaction graph of $f$, which is shown in Fig. 1(d).

Here, edges in an interaction graph are not labeled with additional information pertaining to the activity level values of the tail vertex which allow that edge to have an effect on the dynamics. Thus, an edge $(i, j, \varepsilon)$ may represent several influences of sign $\varepsilon$ from $i$ on $j$, which may differ in strength and depend on the current value of $\alpha_i$. Based on this observation we introduce the following notion.

**Definition 2.3.** *Let $M \subseteq X$ and let $e := (i, j, \varepsilon)$ be an edge in $G(M)$. We call $e$* unique in $M$ *if there exists $t^{ij} \in \{0, \ldots, p_{i-1}\}$ such that $f_j(x) = f_j(x')$ for all $x, x' \in M$ satisfying $x_i, x_i' \in \{0, \ldots, t^{ij}\}$ or $x_i, x_i' \in \{t^{ij}+1, \ldots, p_i\}$, and $x_k = x_k'$ for all $k \neq i$.*

Whether or not edge $e$ has an impact on the dynamical behavior may still be dictated by the values of components other than $\alpha_i$. However, if all component values $x_l$, $l \neq i$, are fixed, the value of $f_j(x)$ solely depends on whether $x_i$ is above or below threshold $t^{ij}$.

By definition, every edge of a circuit influences the behavior of its head vertex at least in some state. When analyzing circuits a stronger property is often useful.

**Definition 2.4.** *Let $C = (e_1, \ldots, e_r)$ be a circuit such that every edge $e_k = (i_k, i_{k+1}, \varepsilon_k)$ is unique in $M \subseteq X$. We call $e_k$* functional in $C$ (with respect to $M$) *if there exists $x \in M$ such that $\overline{x} \in M$ and*

$$f_{i_{k+1}}(x) \leq t^{i_{k+1}, i_{k+2}} < f_{i_{k+1}}(\overline{x}) \quad \text{or} \quad f_{i_{k+1}}(\overline{x}) \leq t^{i_{k+1}, i_{k+2}} < f_{i_{k+1}}(x)\,,$$

*where $\overline{x}$ is the sum of $x$ and the $i_k$-th unit vector.*

For our purposes it is sufficient to introduce the notion of functionality of interactions in circuits for circuits composed of unique edges. For a more general discussion see [18].

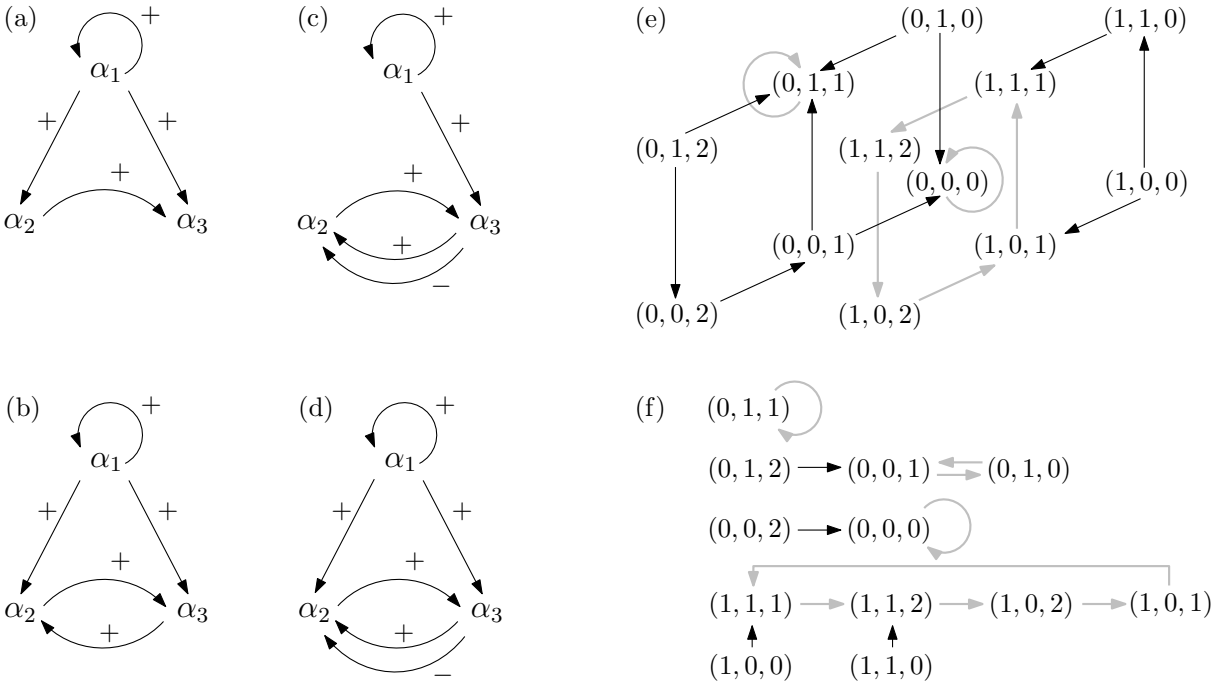



Figure 1: In (a) to (c) local interaction graphs in states (1,0,0), (0,0,0) and (0,0,1), respectively, of the function $f = (f_1, f_2, f_3) : \{0,1\}^2 \times \{0,1,2\} \to \{0,1\}^2 \times \{0,1,2\}$ with $f_1(x) = x_1$, $f_2(x) = 0$ if $x_3 = 2$ and $f_2(x) = x_1 + x_3 - x_1 \cdot x_3$ otherwise, and $f_3(x) = x_1 + x_2$ for $x = (x_1, x_2, x_3) \in \{0,1\}^2 \times \{0,1,2\}$. In (d) the global interaction graph of $f$. In (e) and (f) the asynchronous and synchronous state transition graph, respectively. Heavier gray edges indicate attractors.

## 2.2 Dynamics

The function $f$ determines the behavior of the network. However, there are different possibilities to derive the dynamics of the system. The entirety of the dynamical behavior, in any case, is captured in a *state transition graph* whose paths represent all possible behaviors. The most straightforward approach leads to the following definition.

**Definition 2.5.** *With $S^s := S^s(f)$ we denote the directed graph with vertex set $X$ and edge set $\{(x, f(x)) \mid x \in X\}$. We call $S^s$ the* synchronous state transition graph of $f$.

Here, each state has a unique successor. The underlying assumption concerning the evolution of the system is that all activity level changes indicated by $f$ are executed concurrently. This is a highly simplifying assumption. Changes in activity level may represent very different biological processes that are unlikely to have the exact same duration. Motivated by this observation, we assume that a state differs from its successor in at most one component. Since data on such time delays is often lacking, we have no means to decide which component value should change in a state where multiple component activity level changes are indicated. Therefore, we consider all possibilities and derive a non-deterministic representation of the dynamical behavior. Furthermore, we take into account that although $f$ may indicate an activity level change of absolute value greater than one, the system nevertheless will behave in some sense continuously.

That is, the activity levels of a state and its successor should differ by at most 1.

**Definition 2.6.** *Let $S^a := S^a(f)$ be a directed graph with vertex set $X$. For states $x = (x_1, \dots, x_n)$, $x' = (x'_1, \dots, x'_n) \in X$ there is an edge $x \to x'$ if and only if $x' = f(x) = x$ or $x'_i = x_i + \operatorname{sgn}(f_i(x) - x_i)$ for some $i \in \{1, \dots, n\}$ with $x_i \neq f_i(x)$ and $x'_j = x_j$ for all $j \neq i$. We call $S^a$ the* asynchronous state transition graph of $f$.

To analyze state transition graphs we use, in addition to standard terminology from graph theory such as paths and cycles, the following concepts.

**Definition 2.7.** *Let $S \in \{S^s, S^a\}$. An infinite path $(x^0, x^1, \dots)$ in $S$ is called* trajectory. *A nonempty set of states $D$ is called* trap set *if every trajectory starting in $D$ never leaves $D$. A trap set $A$ is called* attractor *if for all $x^1, x^2 \in A$ there is a path from $x^1$ to $x^2$ in $S$. Attractors of cardinality greater than one are called* cyclic *attractors. A cycle $C := (x^1, \dots, x^r, x^1)$, $r \geq 2$, is called a* trap cycle *if every $x^j$, $j \in \{1, \dots, r\}$, has only one outgoing edge in $S$, i. e., the trajectory starting in $x^1$ is unique. A state $x$ is called* steady state, *if there exists an edge $x \to x$, i. e. if $f(x) = x$.*

In other words, the attractors correspond to the terminal strongly connected components of the graph. In a synchronous state transition graph the trajectory starting from some initial state is unique. In consequence, every attractor is either a fixed point of $f$ or a trap cycle, i. e. a periodic point of $f$. Since the state space is finite, every trajectory leads to an attractor. This is not true for asynchronous state transition graphs. However, it is easy to see that for every state $x$, there exists a trajectory starting in $x$ leading to an attractor. Steady states and trap cycles are attractors, but there may also be attractors of cardinality greater than one which are not trap cycles. Since steady states are fixed points of $f$, they, unlike attractors of cardinality greater than one, coincide in the synchronous and the asynchronous state transition graph.

In Fig. 1 (e) and (f) we see the asynchronous and synchronous state transition graph of the function $f$ defined in the caption. The system has two steady states, (0,1,1) and (0,0,0). The asynchronous state transition graph contains a cyclic attractor consisting of the states $(1, 0, 1)$, $(1, 1, 1)$, $(1, 1, 2)$ and $(1, 0, 2)$. The set containing these states is also an attractor in $S^s$. In the synchronous state transition graph we find a further cyclic attractor, $\{(0, 0, 1), (0, 1, 0)\}$.

We close this section with the following observation. If some coordinate function $f_i$ is constant with value $c$, then $x_i = c$ for every state $x$ in an attractor Similarly, we know the values $x_j$ of every component $j$ such that $f_j$ depends only on values of components whose dynamics are described by constant coordinate functions. That is, we can easily determine the dynamical behavior of such components, which leads to the same fixed values of those components in every attractor of the system. Throughout the remainder of the paper we assume that *no coordinate function of $f$ is constant.* We still allow the system to have input values in the sense of components maintaining their current activity level independent of the values of the other components. They can be modeled with the coordinate function $f_i(x) = x_i$.

## 3 Symbolic Steady States

Analysis of complex network dynamics, in particular of asynchronous state transition graphs, is costly. However, complex networks are often composed of smaller building blocks – modules and motifs (see e. g. [9, 5, 1]). When analyzed in isolation such building blocks often reveal

specific biological functions. The question of interest is whether or not the behavior observed in isolation can be rediscovered in the complex network. Here, the building blocks themselves interact and influence each other. One goal of this paper is to find conditions that allow to infer behavioral properties of the complex system from the dynamics of suitable subnetworks. We introduce the key notions used here in the following two definitions. They have already been established for Boolean functions in [34, 32] and use notation first introduced in [28].

**Definition 3.1.** *For all $i \in \{1,\dots,n\}$ we set $X_i^\theta := \{0,\dots,p_i,\theta\}$ and $X^\theta := X_1^\theta \times \cdots \times X_n^\theta$, where $\theta$ is a symbolic value. We call the elements of $X^\theta$* states*. If no component of a state has value $\theta$, the state is called* regular state*, otherwise it is called* symbolic state*. We denote $J(x) := \{i \in \{1,\dots,n\} \mid x_i = \theta\}$ for all $x \in X^\theta$.*

The value $\theta$ is used to describe uncertainty of a component value. Following this idea, we define a so-called *qualitative value* $|a,b|$ for $a,b \in \{0,\dots,\max_i p_i\}$, $a \le b$ by setting $|a,a| := a$ and $|a,b| := \theta$ if $a < b$. Furthermore, we denote $[x] := \{x' \in X \mid x'_j = x_j \text{ for all } j \notin J(x)\}$ for all $x \in X^\theta$. The set $[x]$ constitutes an affine subspace of the global state space $X$. However, $[x]$ is not necessarily closed with respect to the dynamics of $f$, meaning trajectories starting in $[x]$ may leave $[x]$.

Symbolic notation for logical states, often called schema, has been used before (see e. g. [8, 2, 14, 36]). Schemata are mostly utilized to obtain a compact representation of state space. For example, the different states of a cycle can be represented by a single schema. Schemata representations can also be used as a measure of complexity. In [36], basins of attraction of a system are represented by a set of schemata constructed based on a minimum description principle. The size of the schemata set reflects the complexity of the structure of the basin of attraction in state space.

Although we may encode state sets of relevance for the system's dynamics, the schemata-based representation itself is inherently static, i. e., there is no notion of the dynamical behavior of a schema. In this paper, we want to analyze the dynamics of the system with respect to the set of states represented by a symbolic state, reminiscent of symbolic dynamics of real or complex dynamical systems (see e. g. [30]). However, instead of encoding trajectories in sequences of identifiers of state space subsets such that the system's dynamics is determined by the shift operation on such a sequence, we define a function that allows us to analyze the dynamics of symbolic states on $X^\theta$ and preserves the dynamical behavior derived from $f$.

**Definition 3.2.** *For all $i \in \{1,\dots,n\}$ we define*

$$f_i^{\min} : X^\theta \to X_i^\theta, \;\; f_i^{\min}(x) := \min\{f_i(x') \mid x' \in [x]\}$$

*and*

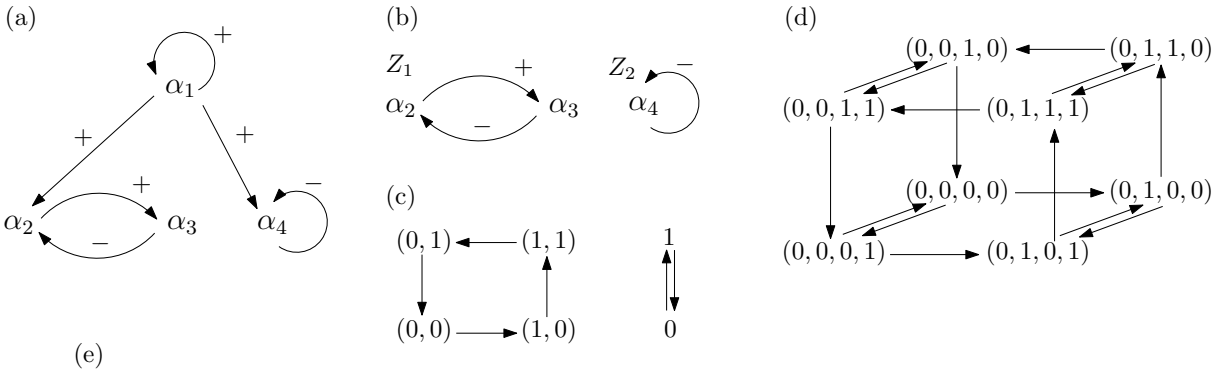
$$f_i^{\max} : X^\theta \to X_i^\theta, \;\; f_i^{\max}(x) := \max\{f_i(x') \mid x' \in [x]\}\,.$$

*Then we define*

$$f^\theta : X^\theta \to X^\theta \quad \text{by} \quad f_i^\theta(x) = |f_i^{\min}(x), f_i^{\max}(x)|\,,\; i \in \{1,\dots,n\}.$$

*We call $x \in X^\theta$ a* steady state *if $f^\theta(x) = x$.*

By definition we have $f^\theta|_X = f$, since regular states are mapped to regular states. If $f_i(x) \in X_i$ for a symbolic state $x$, we can deduce that $f_i(x) = f_i(y)$ for all $y \in [x]$. That is,




(e)

| $f(x) = x$ | attractors (async) | attractors (sync) |
|---|---|---|
| $(0, \theta, \theta, \theta)$ | $\{x \in X \mid x_1 = 0\}$ | $\{(0,0,0,0), (0,1,0,1), (0,1,1,0), (0,0,1,1)\}$ $\{(0,0,0,1), (0,1,0,0), (0,1,1,1), (0,0,1,0)\}$ |
| $(1, 1, 1, \theta)$ | $\{(1,1,1,0), (1,1,1,1)\}$ | $\{(1,1,1,0), (1,1,1,1)\}$ |
| $(2, 1, 1, 1)$ | $\{(2,1,1,1)\}$ | $\{(2,1,1,1)\}$ |

Figure 2: In (a) the global interaction graph of $f = (f_1, f_2, f_3, f_4) : X \to X$, $X := \{0,1,2\} \times \{0,1\}^3$ with $f_1(x) = x_1$, $f_2(x) = 1$ if $x_1 \geq 1$ or $x_3 = 0$, and $f_2(x) = 0$ otherwise, $f_3(x) = x_2$, and $f_4(x) = 1$ if $x_1 = 2$ or $x_4 = 0$, and $f_4(x) = 0$ otherwise. The graph $G^\theta([(0, \theta, \theta, \theta)])$ is shown in (b). The corresponding state transition graphs, and the derived compositional attractor in $S^a$ of $f$ in (c) and (d), respectively. In (e) a table of attractors derived from subgraphs induced by steady states given in the left column.

the information inherent in the regular components of $x$ is sufficient to determine the evolution of the $i$-th component. This can be visualized by looking at the local interaction graph $G([x])$, which then does not contain edges from any $\alpha_j \in J(x)$ to $\alpha_i$.

Since we made the assumption that no coordinate function of $f$ is constant, we know that the state $(\theta, \ldots, \theta)$ is a steady state. In the following we are interested in symbolic steady states $x$ with $J(x) \neq \{1, \ldots, n\}$. The set of components not belonging to $J(x)$ remains dynamically stable regardless of value changes in the symbolic components. This allows us to consider a reduced network for analyzing the system's behavior in the subset $[x]$ of state space without loosing information. The regular components of the symbolic steady state act as a stable or *frozen core*, as was described first by S. Kauffman for random Boolean networks (see [11] for an overview).

To formalize the impact of a symbolic steady state on the dynamics and relate certain structural characteristics, we introduce the following notation for a symbolic steady state $x$. By $G^\theta([x]) = (V^\theta[x], E^\theta[x])$ we denote the (multi-)graph with

$$V^\theta[x] := J(x) \quad \text{and} \quad E^\theta[x] := \{(i, j, \varepsilon) \in G([x]) \mid i, j \in J(x)\}\,.$$

We call a graph $Z = (V_Z, E_Z)$ component of $G^\theta([x])$, if the undirected graph derived from $Z$ is a maximal connected subgraph of the undirected graph derived from $G^\theta([x])$. Figure 2 (b) shows the graph $G^\theta([x])$ for the symbolic steady state $x := (0, \theta, \theta, \theta)$ of the function $f$ defined in the corresponding caption.

**Proposition 3.3.** *Let $x \in X^\theta$ be a symbolic steady state and let $Z$ be a component of $G^\theta([x])$. Then every vertex of $Z$ has at least one predecessor in $Z$. In particular, $Z$ contains a circuit.*

*Proof.* Let $\alpha_i$ be a vertex in $Z$. Since $f_i(x) = x_i = \theta$, we have $f_i^{\min}(x) \neq f_i^{\max}(x)$. Thus, according to the definition, $f_i$ depends on some $\alpha_j \in J(x)$ and we find an edge $(j, i, \varepsilon)$ in $Z$ for some $\varepsilon \in \{+, -\}$. Since the vertex set of $Z$ is finite, there has to be a circuit in $Z$. □

In the next lemma we show that the regular components of a symbolic steady state $x$ stay fixed regardless of value changes in $J(x)$, and that the components of $G^\theta([x])$ act dynamically independent from each other in the state set $[x]$. In particular, $[x]$ is not only a subspace of state space but also a trap set. This has already been shown for Boolean functions in [32] and the proof can be adapted easily.

**Lemma 3.4.** *Let $x \in X^\theta$ be a symbolic steady state, and let $Z_1, \ldots, Z_m$ be the components of $G^\theta([x])$. Consider a union $Z$ of arbitrary components $Z_j$. Let $\tilde{x} \in X^\theta$ such that $\tilde{x}_i = x_i$ for all $i \notin Z$. Then $f_i^\theta(\tilde{x}) = f_i^\theta(x) = x_i = \tilde{x}_i$ for all $i \notin Z$.*

*Proof.* First, let us consider $i \notin J(x)$. Since $x_j = \theta$ for all $j \in Z$, we have $J(\tilde{x}) \subseteq J(x)$. Therefore, $[\tilde{x}] \subseteq [x]$. It follows that $f_i^{\min}(x) \leq f_i^{\min}(\tilde{x}) \leq f_i^{\max}(\tilde{x}) \leq f_i^{\max}(x)$. Since $f_i^\theta(x) = x_i$ is regular, we know $f_i^{\min}(x) = f_i^{\max}(x) = x_i$. Thus, $f_i^{\min}(\tilde{x}) = f_i^{\max}(\tilde{x}) = x_i$ and $f_i^\theta(\tilde{x}) = x_i = \tilde{x}_i$.

Now, let us consider $i \in J(x) \setminus Z$. We need to show that $f_i^\theta(\tilde{x}) = \theta$. Assume $f_i^\theta(\tilde{x}) = c \in X_i$, i. e. $f_i^{\min}(\tilde{x}) = f_i^{\max}(\tilde{x}) = c$. Since $f_i^\theta(x) = \theta$ there exist $y, y' \in [x]$ and $a, b \in X_i$ such that $f_i(y) = a < b = f_i(y')$. We may assume that $a \neq c$, since $a \neq c$ or $b \neq c$. It follows that $y \notin [\tilde{x}] \subseteq [x]$. Since $x$ and $\tilde{x}$ only differ in $Z$-components, there exists $\tilde{y} \in [\tilde{x}]$ with $y_j = \tilde{y}_j$ for all $j \notin Z$. Then we find regular states $y =: y^1, y^2, \ldots, y^k := \tilde{y}$ in $[x]$ such that for each $l < k$ exists $j^l \in Z$ with $|y^l_{j^l} - y^{l+1}_{j^l}| = 1$ and $y^l_m = y^{l+1}_m$ for $m \neq j^l$, and $y^l \notin [\tilde{x}]$ for all $l < k$. Since $f_i(y^k) = c \neq a$, we find $y^l$ such that $f_i(x^l) = a \neq f_i(y^{l+1})$. According to the definition of the local interaction graph, we then find an edge from $j^l$ to $i$ in $G(y^l)$, and thus also in $G([x])$. This is a contradition to $Z$ being a union of components of $G^\theta([x])$, since $j^l \in J(x) \setminus Z$. □

This lemma allows us to focus on the dynamics of the subnetworks represented by the components of $G^\theta([x])$ and to derive the dynamical behavior of the original network in $[x]$ from that, as we will see in the next section.

# 4 Compositional Attractors

Attractors of a complex system can be constructed from subsystem attractors if the subsystems are independent of each other. Such subsystems are of interest both in a mathematical and a biological context (see e. g. [3, 15]). In the following we show that symbolic steady states allow us to identify suitable subsystems.

We need the following notation. Let $x \in X^\theta$ be a symbolic steady state, and let $Z$ be a component of $G^\theta([x])$. Let $k := \operatorname{card} V_Z$ be the cardinality of $V_Z$. We may assume that $V_Z = \{\alpha_{l+1}, \ldots, \alpha_{l+k}\}$ for some $l \in \{0, \ldots, n-1\}$. Set $X^Z := X_{l+1} \times \cdots \times X_{l+k}$. We define

$$f^Z : X^Z \to X^Z\,, \quad f^Z := \pi^Z \circ f^\theta \circ \rho^Z\,,$$

where $\rho^Z : X^Z \to X^\theta$ with $\rho_i^Z(z) = x_i$ for $i \notin Z$ and $\rho_i^Z(z) = z_{i-l}$ for $i \in Z$, and $\pi^Z : X^\theta \to X^Z$ is the projection on the components of $Z$. The function $f^Z$ maps regular states to regular states,

since $Z$ is disjoint from all other components $\alpha_j \in J(x)$ in $G^\theta([x])$ and thus $f_i^\theta(\rho(z)) \in X_i$ for $i \in Z$. Proposition 3.3 ensures that no coordinate function of $f^Z$ is constant. Furthermore, it is easy to see that the global interaction graph $G(f^z)$ is isomorphic to $Z$. We denote the synchronous and asynchronous state transition graph derived from $f^Z$ by $S^s_Z$ and $S^a_Z$, respectively.

In the remainder of the section let $x$ be a symbolic steady state of $f$, and let $Z_1, \ldots, Z_m$ be the components of $G^\theta([x])$. W. l. o. g. we may assume that $Z_1$ contains the vertices $\alpha_1, \ldots, \alpha_{\operatorname{card} Z_1}$, and $V_{Z_i} = \{\alpha_{1+\sum_{j=1}^{i-1} \operatorname{card} Z_j}, \ldots, \alpha_{\sum_{j=1}^{i} \operatorname{card} Z_j}\}$ for all $i \in \{2, \ldots, m\}$, and thus $\{1, \ldots, n\} \setminus J(x) = \{k, \ldots, n\}$ for $k := 1 + \sum_{j=1}^{m} \operatorname{card} Z_j$.

To simplify notation we identify subsets of $X^{Z_1} \times \cdots \times X^{Z_m} \times \{(x_k, \ldots, x_n)\}$ with subsets of $X$.

**Theorem 4.1.** *For all $i \in \{1, \ldots, m\}$ let $A_i$ be an attractor in $S^a_{Z_i}$. Then*

$$A := A_1 \times \cdots \times A_m \times \{(x_k, \ldots, x_n)\}$$

*is an attractor of the asynchronous state transition graph $S^a$ of $f$. Moreover, every attractor in $S^a$ with all its vertices in $[x]$ can be represented in this manner as the Cartesian product of attractors in $S^a_{Z_i}$, $i \in \{1, \ldots, m\}$, and $\{(x_k, \ldots, x_n)\}$.*

*Proof.* We observe that

$$\pi^{Z_j}(f(x')) = \pi^{Z_j}(f^\theta(x')) = \pi^{Z_j}(f^\theta(\rho^{Z_j}(\pi^{Z_j}(x')))) = f^{Z_j}(\pi^{Z_j}(x'))$$

for all $x' \in [x]$ and all $j \in \{1, \ldots, m\}$ according to the definition of $\rho^{Z_j}$, $\pi^{Z_j}$ and Lemma 3.4.

By definition we have $A \subseteq [x]$. We show that $A$ is a trap set in $S^a$ which is strongly connected. Let $a \in A$. If $f(a) = a$ then $A$ is an attractor. Otherwise choose $i \in \{1, \ldots, n\}$ such that $f_i(a) \neq a_i$. Consider $a' \in X$ with $a'_i = a_i + \operatorname{sgn}(f_i(a) - a_i)$ and $a'_j = a_j$ for $j \neq i$. Since $a \in [x]$ we have $f_j(a) = a_j$ for all $j \in \{k, \ldots, n\}$. Thus there is $l \in \{1, \ldots, m\}$ with $i \in Z_l$. Then we choose $k^l$ such that $i = j^l + k^l$ with $j^l := \sum_{j=1}^{l-1} \operatorname{card} Z_j$. We have $a_i \neq f_i(a) = f^{Z_l}_{k^l}(\pi^{Z^l}(a))$. It follows that $a_i + \operatorname{sgn}(f_i(a) - a_i) = (\pi^{Z_l}(a))_{k^l} + \operatorname{sgn}(f^{Z_l}_{k^l}(\pi^{Z_l}(a)) - (\pi^{Z_l}(a))_{k^l})$. Since $A^l$ is a trap set in $S^a_{Z_l}$, it follows that $\pi^{Z_l}(a') \in A_l$, and thus $a' \in A$.

To obtain a path from $a, a' \in A$ in $S^a$ we construct the path componentwise in the attractors $A_i$. More precisely, we exploit the fact that if $\gamma$ is a path in $S^a_{Z_i}$ from $z$ to $z'$ for $z, z' \in X^{Z_i}$, then we find a path $\gamma'$ in $S^a$ from $y$ to $y'$ with $y_j = y'_j$ for all $j \notin Z_i$ which is projected on $\gamma$ by $\pi^{Z_i}$. This is possible since the dynamics in $Z_1, \ldots, Z_m$ do not influence each other according to Lemma 3.4 and since in the asynchronous state transition graph state changes indicated by $f$ are executed componentwise. For a detailed elaboration of this argument in the Boolean case see [32], proof of Theor. 5.6. It follows, that $A$ is strongly connected.

The same reasoning ensures that the projection $\pi^{Z_j}(A)$, $j \in \{1, \ldots, m\}$, of an arbitrary attractor $A$ of $S^a$ is an attractor of $S^a_{Z_j}$. Edges leaving $\pi^{Z_j}(A)$ in $S^a_{Z_j}$ would generate edges leaving $A$ in $S^a$, and paths in $A$ are projected on sequences of states in $X^{Z_j}$ that constitute a path in $S^a_{Z_j}$, if we eliminate all but one consecutive identical states (due to non-injectivity of $\pi^{Z_l}$) in the sequence. □

The proof suggests that the compositional properties inherent in $x$ are not restricted to the construction of attractors. We have shown in [32] in the Boolean case that in fact the subgraph

of $S^a$ with vertex set $[x]$ is the composition of the graphs $S^a_{Z_i}$, $i \in \{1, \ldots, m\}$, and $\{x_k, \ldots, x_n\}$. The reasoning can be adopted for multi-valued functions.

In Fig. 2 (e) we see in the second column of the table attractors derived from the corresponding symbolic steady state in the first column. For the state $(0, \theta, \theta, \theta)$ we obtain two components of $G^\theta([x])$ as shown in Fig. 2 (b). In (c) the corresponding graphs $S^a_{Z_1}$ and $S^a_{Z_2}$ are shown, and in (d) we see the compositional attractor in the state transition graph $S^a$ of $f$.

In [7, 6] the authors consider the case that $G(f)$ is not connected and show how to derive attractors of the synchronous state transition graph from the subnetwork dynamics corresponding to the graph components. Lemma 3.4 allows us to apply their results to the components $Z_1, \ldots, Z_m$ and obtain the following statement concerning the synchronous state transition graph of $f$.

**Theorem 4.2.** *For all $i \in \{1, \ldots, m\}$ let $A_i$ be an attractor in $S^s_{Z_i}$. Then the set*

$$\mathcal{A} := A_1 \times \cdots \times A_m \times \{(x_k, \ldots, x_n)\}$$

*is a union of attractors in $S^s$ such that the cardinality of each attractor $A \in \mathcal{A}$ is the least common multiple of the cardinalities of the attractors $A_i$, $i \in \{1, \ldots, m\}$, and, as a consequence, the number of attractors in $\mathcal{A}$ is*

$$\prod_{j=2}^{m} ((\operatorname{card} A_1 \bullet \operatorname{card} A_2) \bullet \cdots \bullet \operatorname{card} A_{j-1}) \star \operatorname{card} A_j \,,$$

*where $\bullet$ denotes the least common multiple and $\star$ the greatest common divisor operation. Moreover, $\pi^{Z_j}(A)$ is an attractor in $S^s_{Z_j}$ for every attractor $A \subseteq [x]$ in $S^s$ and every $j \in \{1, \ldots, m\}$.*

It is easy, but tedious, to describe the states of a compositional attractor in $S^s$. It is basically a concatenation of the steady states resp. cycles of the subsystems. Since we update all components at once in every step, cycles in $S^s_{Z_i}$ can generate more than one cycle in $S^s$. This is illustrated by the example in Fig. 2. The last column of the table in (e) shows attractors in $S^s$ derived from different symbolic steady states. The composition of the two attractors shown in (c) yield two cycles in the synchronous dynamics.

Comparing the synchronous and the asynchronous case, the differences become apparent when looking at the attractors derived from component attractors of cardinality greater than one. In the synchronous case, composition of component attractors may result in a greater number of attractors, while in the asynchronous case only the attractor size increases. A more detailed description of the relation between cyclic attractors in synchronous and asynchronous dynamics would necessarily include observations on how edges are generated during composition of attractors in the asynchronous case as well as on effects of the gradual activity level change of the asynchronous update.

We close this section with a simple corollary from the two preceding theorems, which for the synchronous case has already been formulated in [6].

**Corollary 4.3.** *For $i \in \{1, \ldots, m\}$ let $S^\delta_{Z_i}$, $\delta \in \{a, s\}$, contain $N_i$ attractors with cardinalities $L_{ij}$, $j \in \{1, \ldots, N_i\}$. Set $I := I_1 \times \cdots \times I_m$ with $I_l := \{1, \ldots, N_l\}$.*

- *In the asynchronous state transition graph of $f$, the number of attractors with vertices in $[x]$ is $\prod_{j=1}^m N_j$ and the maximal attractor cardinality in $[x]$ is $\max_{(k_1,\ldots,k_m)\in I} \prod_{j=1}^m L_{jk_j}$.*

- *In the synchronous state transition graph of $f$, the number of attractors with vertices in $[x]$ is $\sum_{(k_1,\ldots,k_m)\in I}\prod_{j=2}^{m}((L_{1k_1}\bullet L_{2k_2})\bullet\cdots\bullet L_{j-1k_{j-1}})\star L_{jk_j}$, and the maximal attractor cardinality in $[x]$ is $\max_{(k_1,\ldots,k_m)\in I}((L_{1k_1}\bullet L_{2k_2})\bullet\cdots\bullet L_{mk_m})$.*

# 5 Circuits and Attractors

In the preceding section we constructed attractors in the dynamics of $f$ from attractors of functions $f^{Z_i}$ associated with certain subnetworks. The same reasoning allows us to formulate more general relations between structural characteristics of the subnetworks and the dynamics of $f$. We focus in this section on the impact of feedback circuits in the structure on number and size of attractors of $f$.

For multi-valued discrete functions with asynchronous update it was shown in [27] that the existence of a positive circuit in the global interaction graph (and even in certain local interaction graphs) is a necessary condition for the existence of two attractors in the asynchronous state transition graph. The existence of a negative circuit in the global interaction graph is necessary for the existence of a cyclic attractor (see [25]). Obviously, the result on positive circuits also holds in the synchronous case if we specify the attractors to be steady states. However, simple examples show that the second result is false in the synchronous case.

We now focus on functions whose global interaction graph contains only one circuit.

**Lemma 5.1.** *Assume the global interaction graph $G(f)$ contains only one circuit $C$, and let $V_C$ be the set of vertices visited by $C$. Then $f_i(x) = f_i(x')$ for all $i \in V_C$ and $x, x' \in X$ with $x_j = x'_j$ for all $j \in V_C$.*

*Proof.* Assume we find $i \in V_C$ and $x, x' \in X$ such that $x_j = x'_j$ for all $j \in V_C$ and $f_i(x) \neq f_i(x')$. Then there exist $y, y' \in X$ and $k \notin V_C$ such that $y_j = y'_j$ for all $j \neq k$, $|y_k - y'_k| = 1$ and $f_i(y) \neq f_i(y'_k)$ (see proof of Lemma 3.4). It follows that $G(f)$ contains an edge from $k$ to $i$. Since we always assume that no coordinate function is constant, every vertex in $G(f)$ has a predecessor. Since the set of vertices is finite, we then find a circuit other than $C$ in $G(f)$ which is a contradiction. □

In [22] it is shown that Boolean functions associated with isolated circuits always display a characteristic behavior depending on their sign, both in the synchronous and the asynchronous case. We use this result to prove the following lemma.

**Lemma 5.2.** *Assume the global interaction graph $G(f)$ contains only one circuit $C$ whose edges are unique in $X$ and functional in $C$. If $C$ is a positive circuit, there are at least two attractors in $S^s$ as well as in $S^a$. If $C$ is negative, there exists a cyclic attractor in $S^s$ and in $S^a$.*

*Proof.* Let us assume that $C$ corresponds to the vertex sequence $(\alpha_1, \ldots, \alpha_r)$.

Set $X_C := X_1 \times \cdots \times X_r$ and let $z_i \in X_i$ for $i \in \{r+1, \ldots, n\}$. We set

$$f^C : X^C \to X^C, \quad f^C := \pi^C \circ f \circ \rho^C,$$

where $\rho^C : X^C \to X, y \mapsto (y_1, \ldots, y_r, z_{r+1}, \ldots, z_n)$ and $\pi^C$ is the projection on the components of $C$. According to Lemma 5.1, we have $f_i^C(\pi^C(x)) = \pi^C(f(x))$ for all $x \in X$. It follows that $G(f^C) = C$, i. e., values of coordinate function $f_i^C$ depend only on the value of the predecessor in $C$. Since furthermore all edges of $C$ are unique (see Def. 2.3), we find for all $i \in \{1, \ldots, r\}$ values

$t^{i,i+1} \in \{0,\dots,p_i-1\}$ such that $f^C_{i+1}(y) = f^C_{i+1}(y')$ for all $y, y' \in X^C$ with either $y_i, y'_i \leq t^{i,i+1}$ or $y_i, y'_i > t^{i,i+1}$, indices taken modulo $r$. In addition, there exist $y, y' \in X^C$ with either $y_i \leq t^{i,i+1} < y'_i$ or $y'_i \leq t^{i,i+1} < y_i$ such that $f^C_{i+1}(y) \leq t^{i+1,i+2} < f_{i+1}(y')$, since the edge from $\alpha_i$ to $\alpha_{i+1}$ is functional in $C$. This allows us to define a Boolean function

$$f^{\mathcal{B}} : \{0,1\}^r \to \{0,1\}^r\,, \quad f^{\mathcal{B}} := \pi^t \circ f^C \circ \rho^t$$

with $\rho^t : \{0,1\}^r \to X^C$, $\rho^t_j(0) = c_j$, $\rho^t_j(1) = c'_j$ for arbitrary but fixed $c_j \in \{0,\dots,t^{j,j+1}\}$, $c'_j \in \{t^{j,j+1}+1,\dots,p_j\}$, and $\pi^t : X^C \to \{0,1\}^r$, $\pi^t_j(y) = 0$ if $y_j \leq t^{j,j+1}$ and $\pi^t_j(y) = 1$ if $y_j > t^{j,j+1}$. Note that $f^{\mathcal{B}}$ does not depend on the choice of $c_j$, $c'_j$. It is easy to see that $G(f^{\mathcal{B}})$ coincides with $C$.

In [22] it was shown that $f^{\mathcal{B}}$ has at least two fixed points, if $C$ is positive. Let $b$ be a fixed point of $f^{\mathcal{B}}$. Choose $y' \in (\pi^t)^{-1}(b)$ and set $y_i := f^C_i(y')$ for all $i \in \{1,\dots,r\}$. Since $b$ is a fixed point, we can deduce that $y = (y_1,\dots,y_r) = f^C(y') \in (\pi^t)^{-1}(b)$. It follows from the definition of $\pi^t$ that for all $i \in \{1,\dots,r\}$ either $y_i, y'_i \leq t^{i,i+1}$ or $y_i, y'_i > t^{i,i+1}$. The uniqueness condition then yields $f^C_i(y) = f^C_i(y') = y_i$. Thus $y$ is a fixed point of $f^C$.

We then find a trajectory in $S^\delta$, $\delta \in \{a,s\}$, starting in $y$ and leading to an attractor $A$. According to Lemma 5.1 we then have $f_i(a) = y_i$ for all $i \in \{1,\dots,r\}$ and every $a \in A$. Since we have two different fixed points of $f^C$, if $C$ is positive, we also find two different attractors in $S^\delta$.

If $C$ is negative, then $f^{\mathcal{B}}$ does not have a fixed point (see again [22]). It follows from the definition of $f^{\mathcal{B}}$, in particular from its independence of the choice of $c_j$ and $c'_j$, that $f^C$ does not have a fixed point either. Thus, the synchronous as well as the asynchronous state transition graph of $f^C$ contain a cyclic attractor. Lemma 5.1 then again yields the existence of a cyclic attractor in the asynchronous and synchronous state transition graph of $f$. □

The uniqueness of edges is exploited in the proof to obtain a suitable projection on the Boolean case. However, it seems likely that the statement remains true when dropping the condition. A finer partition of the range of a component corresponding to multiple thresholds could clarify the situation.

By applying the above lemma and Theorems 4.1 and 4.2 (or Cor. 4.3) we immediately obtain the following theorem.

**Theorem 5.3.** *Let $x$ be a symbolic steady state. Assume a component $Z$ of $G^\theta([x])$ contains only one circuit $C$ and that all edges of $C$ are unique in $[x]$ and functional in $C$ with respect to $[x]$. If $C$ is positive, then there exist at least two attractors in $S^a$ as well as in $S^s$. If $C$ is negative, then there is a cyclic attractor in both $S^a$ and $S^s$.*

The theorem basically states that circuits embedded in complex networks imprint some of the characteristics they show in isolation on the whole network under certain conditions. The statement does not hold if the circuit $C$ is not the only circuit in $Z$ (see Boolean examples in [34, 32]). Furthermore, the hypothesis given in the last theorem is a sufficient but not a necessary condition for the existence of multiple resp. cyclic attractors. This is illustrated by an example given in [34], Fig. 4.

# 6 Determining symbolic steady states

As we have seen, symbolic steady states can be very useful in the analysis of the dynamics. In this section, we address the problem of determining symbolic steady states of a given system.

## 6.1 Seeds

In the following we are interested in states that have component values that stay fixed under iteration.

**Definition 6.1.** *We call a state $s \in X^\theta$* seed of $f$ *if $(f^\theta)_i^k(s) = s_i$ for all $i \in \{1,\dots,n\} \setminus J(s)$ and $k \in \mathbb{N}$, where $(f^\theta)^k$ denotes the $k$-th iterate of $f^\theta$.*

By definition, the state $(\theta,\theta,\dots,\theta)$ is a seed. We have mentioned in Sect. 3 that it is also a steady state. In general, a seed is not a steady state, however, we can easily derive a steady state from a seed.

**Lemma 6.2.** *Let $s \in X^\theta$ be a seed of $f$. Then $((f^\theta)^k(s))_{k\in\mathbb{N}}$ converges with respect to the discrete metric on $X^\theta$ to a symbolic steady state, which we call the* symbolic steady state generated by *$s$.*

*Proof.* If $J(s) = \{1,\dots,n\}$ then $s$ is a steady state as mentioned above. So let us assume that $\{1,\dots,n\} \setminus J(s)$ is not empty. We set $s^0 := s$ and $s^l := f^\theta(s^{l-1})$ for all $l \in \mathbb{N}$, and show by induction: for all $l \in \mathbb{N}$ and for all $i \in \{1,\dots,n\}$, if $s_i^l \neq \theta$, then $s_i^l = s_i^m$ for all $m \geq l$. This is true for $l = 0$ since $s$ is a seed.

Let us now assume the hypothesis holds for $l \in \mathbb{N}$. It follows that $[s^m] \subseteq [s^l]$ for all $m \geq l$. Let $i \in \{1,\dots,n\}$ such that $s_i^{l+1} \neq \theta$. Then

$$s_i^{l+1} = f_i^{\min}(s^l) \leq f_i^{\min}(s^m) \leq f_i^{\max}(s^m) \leq f_i^{\max}(s^l) = s_i^{l+1}\,.$$

It follows that $s_i^{m+1} = f_i^{\min}(s^m) = f_i^{\max}(s^m) = s - i^{l+1}$ for all $m \geq l$, which proves our claim. The statement of the lemma follows immediately. □

The proof shows that we reach a symbolic steady state after at most $\operatorname{card} J(s)$ iteration steps. Moreover, it shows that a trap set contained in $[s]$ but not in $[x]$, where $x$ is the symbolic steady state generated by $s$, does not constitute a strongly connected component of either the synchronous or the asynchronous state transition graph. This leads to the following statement.

**Proposition 6.3.** *Let $s \in x^\theta$ be a seed of $f$, and let $x \in X^\theta$ be the symbolic steady state generated by $s$. Then for every attractor $A \subseteq [s]$ in $S \in \{S^a, S^s\}$ holds $A \subseteq [x]$.*

In consequence, all attractors in $[s]$ in the asynchronous resp. the synchronous state transition graph can be derived from $x$ as described in Theorem 4.1 resp. 4.2. In particular, we can find all attractors of the system if we consider a set of seeds $\{s^1,\dots,s^k\}$ such that $X = \bigcup_{i=1}^k [s^i]$.

We illustrate the results using the function $f$ given in Fig. 2. The state $(1,\theta,\theta,\theta)$ is a seed of $f$. After two iteration steps we obtain the symbolic steady state $(1,1,1,\theta) = f^\theta((1,1,\theta,\theta)) = f^\theta(f^\theta((1,\theta,\theta,\theta)))$. It can easily be checked that the attractors given in Fig. 2 (e) are all attractors of the system. We then see that the attractors derived from $(1,1,1,\theta)$ are all the attractors in $[(1,\theta,\theta,\theta)]$.

## 6.2 Forcing structures

We have seen that we can use seeds to determine symbolic steady states, but we still need methods to find seeds. In the following, we focus on exploiting characteristics of the function describing the system as well as of the interaction graph. We start by introducing the notion of canalyzing functions, which is well-known in the Boolean setting, appearing in several different contexts [11, 10].

**Definition 6.4.** *Let $k \in \{1,\dots,n\}$. A function $g : X \to X_k$ is called* canalyzing *if there exist $i \in \{1,\dots,n\}$, $c \in X_i$ and $c' \in X_k$ such that $g(x) = c'$ for all $x \in X$ with $x_i = c$. The $i$-th component is called* canalyzing component*, the value $c$ is called* canalyzing value (of the canalyzing component $i$) *and $c'$ is called* canalyzed value*.*

Already in the Boolean case it is possible for a function to have more than one canalyzing value. Consider for example a projection of a $k$-tuple to the first component. Then, the first component is canalyzing and every possible value of the first component is a canalyzing as well as a canalyzed value.

Recall that we excluded constant coordinate functions. Then it is easy to see that there is an edge from $i$ to $j$ in the global interaction graph, if the $j$-th coordinate function of $f$ is canalyzing with canalyzing component $i$. If canalyzing functions interlock along such edges such that the canalyzing and canalyzed values match, then canalyzing values percolate through the network [11]. To make this statement more precise we use the following definitions.

**Definition 6.5.** *A pair $(i,j) \in V^2$ is called* forcing connection from $i$ to $j$ *if both $f_i$ and $f_j$ are canalyzing functions, $f_j$ with canalyzing component $i$, and a canalyzed value of $f_i$ is a canalyzing value of component $i$ of $f_j$.*

*Let $W \subseteq V^2$ and set $V' := \{l \in V \mid \exists k : (l,k) \in W \vee (k,l) \in W\}$. The set $W$ is called* forcing structure *if there exist values $c_k \in X_k$, $k \in V'$, such that for every $(i,j) \in W$ the value $c_i$ is a canalyzed value of $f_i$ and a canalyzing value of component $i$ of $f_j$. The multiset of values $c_k$, $k \in V'$ is called* assignment of W*. We denote the unsigned directed graph with vertex set $V'$ and edge set $W$ by $G_W$.*

In general, the assignment of $W$ is not unique. Clearly, every subset of $W$ is also a forcing structure and every $(i,j) \in W$ is a forcing connection.. Every element of $W$ corresponds to an edge in $G(f)$. More precisely, $G_W$ is a subgraph of the graph obtained by dropping the edge signs of the edges of $G(f)$. The properties of canalyzing functions and forcing structures give rise to the following dynamical property. If we consider a state $x$ with $x_j = c_j$ for some $j \in V'$ and some assignment $\{c_i \in X_i \mid i \in V'\}$ of $W$, then $f_k(x) = c_k$ for all $k \in V'$ with $(j,k) \in W$. That is, in a first iteration step, all successors of $j$ in $W$ adopt the corresponding assignment values, in a second iteration step all their successors in $W$ adopt the corresponding assignment values, and so on. This observation immediately leads to the following statement.

**Proposition 6.6.** *Let $W$ be a forcing structure with assignment $\{c_i \in X_i \mid i \in V'\}$. Then, for every trajectory $(x^0, x^1, \dots)$ in $S^s$, we have $x^k_l = c_l$ for all $l$ with $(i,l) \in W$ and $x^{k-1}_i = c_i$.*

*Let $x^0 \in X$ with $x^0_j = c_j$ for some $j \in V'$, and let $(j = j^0, j^1, \dots, j^m)$ be a path in $G_W$. Then there exists a path $(x^0, x^1, \dots, x^{m'})$, $m' \in \{0,1,\dots,m\}$, in $S^a$ such that for all $k \in \{0,1,\dots,m\}$ there is $i_k \in \{0,1,\dots,m'\}$ with $x^{i_k}_k = c_k$, and $i_{k-1} \le i_k$ for all $k \in \{1,\dots,m\}$.*

The proposition shows that in the synchronous case the influence of canalyzing functions on the dynamics is straightforward. In the asynchronous case we have to keep in mind that the system is non-deterministic and thus the influence of canalyzing values may not be seen along every path in the state transition graph.

In general, the effect of the canalyzing values of a forcing structure on the dynamics is transient, since components outside of the forcing structure may negate it. However, if $G_W$ includes a circuit then outside influences can be neglected as was already observed in [21].

**Definition 6.7.** *A forcing structure $W$ such that $G_W$ is a circuit is called* self-freezing circuit*.*

A circuit in $G_W$ corresponds to a circuit in $G(f)$ but we have no information on the sign of the circuit in $G(f)$. In general, it can be positive or negative. A clearer understanding of the relation between self-freezing circuits and the corresponding circuits in $G(f)$ should be possible if one considers the notion of circuit functionality.

The importance of self-freezing circuits for the dynamics is rooted in the following observation.

**Lemma 6.8.** *Let $W$ be a self-freezing circuit with assignment $\{c_i \in X_i \mid i \in V'\}$. Then $s \in X^\theta$ with $s_i := c_i$ for all $i \in V'$ and $s_j := \theta$ for $j \in V \setminus V'$ is a seed.*

*Proof.* Let $i \in V'$. Then there exists $j \in V'$ such that $(j, i) \in W$. Since $f_i$ is canalyzing with canalyzing component $j$, canalyzing value $c_j$ and canalyzed value $c_i$, we have $f_i(x) = c_i$ for all $x \in X$ with $x_j = c_j$. It follows that $f_i^\theta(s) = c_i$. So, $f_l^\theta(s) = c_l$ for all $l \in V' = \{1, \ldots, n\} \setminus J(s)$. Inductive reasoning then shows $f^\theta(s)_l^k(s) = c_l = s_l$ for all $l \notin J(s)$ and $k \in \mathbb{N}$. □

The seed derived from the self-freezing circuit in turn generates a symbolic steady state as shown in Sect. 6.1. Summarizing the observations up to this point we obtain the following statement.

**Theorem 6.9.** *Let $W$ be a self-freezing circuit with assignment $\{c_i \in X_i \mid i \in V'\}$, let $s \in X^\theta$ be a seed derived from $W$, and let $x \in X^\theta$ be the symbolic steady state it generates. Then all attractors in $[s]$ in the asynchronous resp. the synchronous state transition graph can be derived from $x$ as described in Theorem 4.1 resp. 4.2.*

Self-freezing circuits provide a high degree of stability. Perturbations outside of the circuit do not influence the values of the seed derived from the self-freezing circuit. Moreover, the set of attractors reachable from a state obtained by such a perturbation is the set of attractors derived from the symbolic steady state generated by the seed according to Theor. 6.9. This is true in the synchronous as well as the asynchronous case.

Perturbation of a circuit component may have an effect. However, it is easy to see that the number of circuit components adopting the corresponding assignment value cannot decrease when iterating the perturbed state, since the predecessor of the perturbed component forces its value back to the canalyzing value and only the successor of the perturbed component may adopt a value other than that of the assignment. If outside influences prevent the change of the successor component value at some point, then the component circuit values again remain fixed to their assignment value under iteration. In particular in the Boolean setting this outcome is very likely [21].

The preceding results provide us with a method to find seeds by determining the circuits of $G(f)$ and checking wether the corresponding unsigned circuits are self-freezing. Since circuits

play an important role in the dynamics of regulatory networks, determining and analyzing circuits in a given network is often already implemented in software designed for analyzing logical models, as e. g. GINsim [16]. The complexity of the problem to decide whether a circuit is self-freezing depends on the length of the circuit and on the complexity of the coordinate functions of the circuit components. In some cases the task is very easy. Consider for example the system introduced in Fig. 2. We immediately see that the loop $(\alpha_1, \alpha_1)$ is self-freezing since $f_1$ is the projection on the first component. In this case, it is possible to derive three different seeds from the loop, namely $(0, \theta, \theta, \theta)$, $(1, \theta, \theta, \theta)$ and $(2, \theta, \theta, \theta)$. The symbolic steady states generated by the seeds are $(0, \theta, \theta, \theta)$, $(1, 1, 1, \theta)$ and $(2, 1, 1, 1)$.

In the example above, all attractors of the system can be derived from the symbolic steady states generated by the seeds corresponding to the self-freezing circuit $(\alpha_1, \alpha_1)$. In general, it is not possible to find a self-freezing circuit with that property. In the next section we explore methods to find a set of seeds that allows us to recover all attractors of the system.

## 6.3 Input networks

In Lemma 3.4 we have shown that we can analyze subnetworks derived from the local interaction graph corresponding to a symbolic steady state independently. This is possible because they do not influence each others behavior in a subspace of state space, which is reflected in the fact that there are no edges between them in the corresponding local interaction graph. The same reasoning applies when considering subgraphs $G' = (V', E')$ of $G(f)$ such that there are no edges from vertices in $V \setminus V'$ to vertices in $V'$ in $G(f)$. We can then define a function

$$f^{G'} := \pi^{G'} \circ f \circ \rho^{G'} : X^{G'} \to X^{G'}\,, \quad X^{G'} := \prod_{i \in V'} X_i\,,$$

analogously to the function $f^Z$ introduced in the beginning of Sect. 4. Here, the coordinate functions $\rho_i^{G'}$ are the projections on the $i$-th component for all $i \in V'$, while we set $\rho_i^{G'}(z) = c_i$ with $c_i \in X_i$ arbitrary but fixed for all $i \notin V'$, and $\pi^{G'}$ is again the projection on the components of $V'$.

**Definition 6.10.** *Let $V' \subseteq V$. The subgraph $G' = (V', E')$ of $G(f)$ induced by $V'$ is called* input network of $f$ *if*

1. *there are no edges from vertices in $V \setminus V'$ to vertices in $V'$ in $G(f)$,*
2. *the only attractors of the synchronous and asynchronous state transition graphs of the function $f^{G'}$ are fixed points of $f^{G'}$.*

The definition immediately yields that $f^{G'}$ has at least one fixed point. The notion of input network generalizes the idea of *input vertex*, i. e., a vertex $\alpha_i$ such that $f_i(x) = x_i$, introduced in [32]. An input network does not necessarily have to be connected. If the graph $G'$ consists of several components, then the fixed points of each component can be calculated independently. The fixed points of $f^{G'}$ are then composed of the fixed points of the components. This observation allows for a straightforward approach to networks with input vertices, since the set of fixed point associated with an input vertex coincides with the range of the vertex.

Let us consider the system presented in Fig. 2, and, in particular, the subgraph $G' := (\{\alpha_1\}, \{(\alpha_1, \alpha_1)\})$ of $G(f)$. The only edge leading to $\alpha_1$ is the one included in $G'$. The function

$f^{G'} : \{0,1,2\} \to \{0,1,2\}$ is the identity ($\alpha_1$ is an input vertex). The synchronous as well as the asynchronous state transition graph of $f^{G'}$ consist of three loops originating in the three states of the state space $\{0,1,2\}$, i. e., every state is a fixed point of $f^{G'}$, and $G'$ is an input network.

The next statement follows directly from the lack of edges from $V \setminus V'$ to $V'$ in $G(f)$ and the definition of edges in Def. 2.1.

**Proposition 6.11.** *Let $G' = (V', E')$ be an input network of $f$, and let $y \in X^{G'}$ be a fixed point of $f^{G'}$. Then $s \in X^{\theta}$ with $s_i := y_i$ for all $i \in V'$ and $s_i := \theta$ for all $i \in V \setminus V'$ is a seed.*

Again, we can use the seeds corresponding to $G'$ to generate symbolic steady states and determine attractors. We obtain all attractors of $f$ if we consider the set of seeds corresponding to the set of fixed points of $f^{G'}$.

**Theorem 6.12.** *Let $G' = (V', E')$ be an input network of $f$. Then all attractors in the asynchronous resp. the synchronous state transition graph can be derived from the symbolic steady states generated by the seeds of $G'$ as described in Theorem 4.1 resp. 4.2.*

*Proof.* Let $A$ be an attractor of $f$. It suffices to show that there exists a fixed point $y \in X^{G'}$ such that $a_i = y_i$ for all $i \in V'$, $a \in A$. Then $A \subset [s]$, where $s$ is the seed derived from the fixed point $y$, and the theorem follows with Prop. 6.3.

It is easy to see in the asynchronous case (compare the second part of the proof of Theorem 4.1) and even more obvious in the synchronous case that $\pi^{G'}(A)$ is an attractor of $f^{G'}$. Since $G'$ is an input network, there is a fixed point $y \in X^{G'}$ such that $\pi^{G'}(A) = \{y\}$. The assertion follows. □

Lemma 6.2 and Theorem 6.12 yield a simple strategy for the analysis of systems with input networks. We first determine the fixed points of the input network. As mentioned above, this is particularly easy for input vertices, which can be used to model input in biological networks, e. g. in signal transduction networks. We then obtain symbolic steady states from the corresponding seeds using the simple iteration procedure described in Lemma 6.2. Implementation of this procedure is straightforward. The resulting subsystems can then be used to determine the attractors, and to obtain a detailed understanding of the link between structural and dynamical characteristics.

# 7 T helper cell differentiation

In [13] L. Mendoza proposes a model for a control network regulating differentiation of T helper cells (Th cells), which play an important role in the vertebrate immune system. There exist different types of Th cells involved in different immune responses, namely Th1 and Th2 cells, that originate from a common percursor. Mendoza's model consists of 17 components, represented mostly by Boolean variables, but also by variables with three possible values. The system is described by the logical rules given in Table 1. The global interaction graph is given in Fig. 3. Note that the logical functions associated with the vertices IFN-$\beta$, IL-12 and Il-18 listed in Table 1 differ slightly from those given in [13]. Mendoza models all three components with constant functions with value zero, which represents wild type cells in a natural environment. When considering specific artificial environmental conditions, other constant functions are considered. We model the three components as input vertices. In consequence, different

| | | |
|---|---|---|
| IFN-$\beta$: | $X_1 = \{0,1\}$, | $f_1(x) = x_1$ |
| IL-12: | $X_2 = \{0,1\}$, | $f_2(x) = x_2$ |
| IL-18: | $X_3 = \{0,1\}$, | $f_3(x) = x_3$ |
| IFN-$\beta$R: | $X_4 = \{0,1\}$, | $f_4(x) = x_1$ |
| IFN-$\gamma$: | $X_5 = \{0,1,2\}$, | $f_5(x) = 1$ if $(x_{16} = 1 \wedge \neg(x_{14} = 1 \wedge x_{15} = 1)) \vee$ $(x_{14} = 1 \wedge x_{15} = x_{16} = 0)$, $f_5(x) = 2$ if $x_{16} = 2 \vee x_{14} = x_{15} = 1$, and $f_5(x) = 0$ otherwise |
| IL-4: | $X_6 = \{0,1\}$, | $f_6(x) = 1$ if $x_{12} = 0 \wedge x_{17} = 1$, and $f_6(x) = 0$ otherwise |
| IFN-$\gamma$R: | $X_7 = \{0,1,2\}$, | $f_7(x) = 1$ if $x_5 = 1 \vee (x_5 = 2 \wedge x_{11} = 1)$, $f_7(x) = 2$ if $x_5 = 2 \wedge x_{11} = 0$, and $f_7(x) = 0$ otherwise |
| IL-4R: | $X_8 = \{0,1\}$, | $f_8(x) = x_6 \wedge \neg x_{11}$ |
| IL-12R: | $X_9 = \{0,1\}$, | $f_9(x) = x_2 \wedge \neg x_{13}$ |
| IL-18R: | $X_{10} = \{0,1\}$, | $f_{10}(x) = x_3 \wedge \neg x_{13}$ |
| SOCS-1: | $X_{11} = \{0,1\}$, | $f_{11}(x) = 1$ if $x_{12} \geq 1 \vee x_{16} \geq 1$, and $f_{11}(x) = 0$ otherwise |
| STAT-1: | $X_{12} = \{0,1,2\}$, | $f_{12}(x) = 1$ if $(x_4 = 1 \wedge x_7 = 0) \vee x_7 = 1$, $f_{12}(x) = 2$ if $x_7 = 2$, and $f_{12}(x) = 0$ otherwise |
| STAT-6: | $X_{13} = \{0,1\}$, | $f_{13}(x) = x_8$ |
| STAT-4: | $X_{14} = \{0,1\}$, | $f_4(x) = x_9 \wedge \neg x_{17}$ |
| IRAK: | $X_{15} = \{0,1\}$, | $f_{15}(x) = x_{10}$ |
| T-bet: | $X_{16} = \{0,1,2\}$, | $f_{16}(x) = 1$ if $(x_{17} = 0 \wedge ((x_{12} = 1 \wedge x_{16} \leq 1) \vee$ $(x_{12} \leq 1 \wedge x_{16} = 1))) \vee (x_{17} = x_{16} = x_{12} = 1)$ $f_{16}(x) = 2$ if $(x_{17} = 0 \wedge (x_{12} = 2 \vee x_{16} = 2)) \vee$ $(x_{17} = x_{12} = 1 \wedge x_{16} = 2)$, $f_{16}(x) = 0$ otherwise |
| GATA-3: | $X_{17} = \{0,1\}$, | $f_{17}(x) = 1$ if $x_{13} = 1 \wedge x_{16} = 0$, and $f_{17}(x) = 0$ otherwise |

Table 1: Coordinate functions and ranges for the components of the Th cell network.

environmental conditions correspond to different input values and thus to different subgraphs of state space. In particular, Mendoza's original model corresponds to the situation where all input values are zero.

The vertices $\alpha_1 =$IFN-$\beta$, $\alpha_2 =$IL-12 and $\alpha_3 =$Il-18 represent the only input vertices of the network. The graph $G'$ with vertex set $\{\alpha_1, \alpha_2, \alpha_3\}$ and edge set $\{(\alpha_i, \alpha_i, +) \mid i \in \{1,2,3\}\}$ is an input network, since every state in $X_1 \times X_2 \times X_3 = \{0,1\}^3$ is a fixed point of

$$f^{G'} : \{0,1\}^3 \to \{0,1\}^3, \ (x_1, x_2, x_3) \mapsto (x_1, x_2, x_3).$$

It follows that every $x \in X = \prod_{i=1}^{17} X_i$ with $x_i \in \{0,1\}$ for $i \in \{1,2,3\}$ and $x_i = \theta$ for $i \in \{4, \ldots, 17\}$ is a seed of the system.

Let us analyze the system for input values $x_i = 0$, $i \in \{1,2,3\}$, corresponding to the wild type model in [13]. We start by determining the symbolic steady state generated by the seed

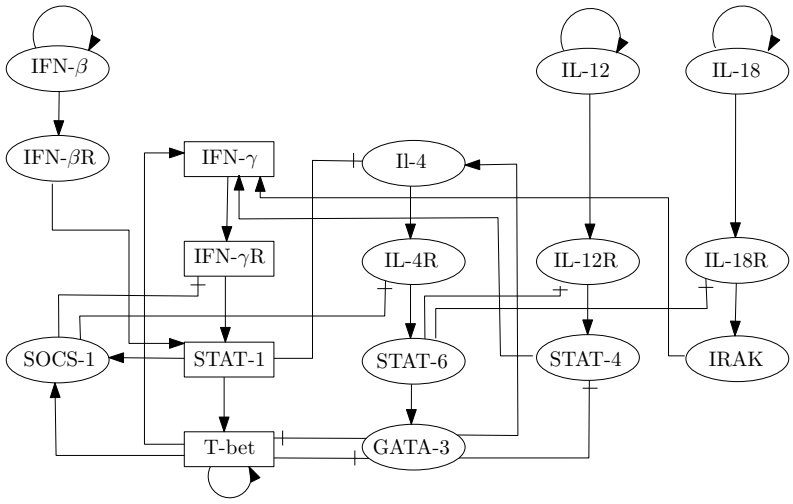



Figure 3: Global interaction graph of the Th cell differentiation network introduced in [13]. Arrows represent activation, crossed lines represent inhibition.

$s^0 := (0,0,0,\theta,\ldots,\theta)$. Iterating $s$ yields

$$\begin{aligned} s^1 :=\ & f^\theta(s^0) = \quad (0,0,0,0,\theta,\theta,\theta,\theta,0,0,\theta,\theta,\theta,\theta,\theta,\theta,\theta,\theta),\\ s^2 :=\ & f(s^1) = \quad (0,0,0,0,\theta,\theta,\theta,\theta,0,0,\theta,\theta,\theta,0,0,\theta,\theta),\\ & f(s^2) = \quad s^2. \end{aligned}$$

The symbolic steady state $s^2$ has eight regular components. The graph $G^\theta[s^2]$ is shown in Fig. 4 (a). The state space of the original system consists of 663552 states. Fixing the values of the components $\alpha_1$, $\alpha_2$ and $\alpha_3$ reduces the size to 82944 states. The state space of $f^{G^\theta[s^2]}$ contains only 2592 states. Analyzing the reduced model still renders all attractors of the system for input values $x_i = 0$, $i \in \{1,2,3\}$ according to Theorem 4.1 resp. 4.2. The system has four attractors, all of them steady states, namely

$$(0,0,0,0,0,0,0,0,0),\ \ (1,0,1,0,1,1,0,1,0),\ \ (2,0,1,0,1,1,0,2,0),\ \ (0,1,0,1,0,0,1,0,1)$$

in the lower-dimensional state space $X_5 \times X_6 \times X_7 \times X_8 \times X_{11} \times X_{12} \times X_{13} \times X_{16} \times X_{17}$. Expanding the states with the regular components of $s^2$ then yields the four steady states of the original system. They are in complete agreement with the results in [13], where it is also mentioned that each of the discovered steady states has a clear biological interpretation.

Instead of calculating the attractors for $f^{G^\theta[s^2]}$, we could also use the idea of self-freezing circuits to simplify the system even further. The graph shown in Fig. 4 (a) contains 10 different circuits, not all of them self-freezing. Let us look at two examples of self-freezing circuits of $f^{G^\theta[s^2]}$.

The components

$$\alpha_6 = \text{IL-4},\ \ \alpha_8 = \text{IL-4R},\ \ \alpha_{13} = \text{STAT-6 and}\ \ \alpha_{17} = \text{GATA-3}$$

constitute a circuit, and it is easy to see that the corresponding functions $f_i$ are canalyzing. An assignment of the circuit is given by the values $c_i := 0$ ,$i \in \{6,8,13,17\}$, i. e.,

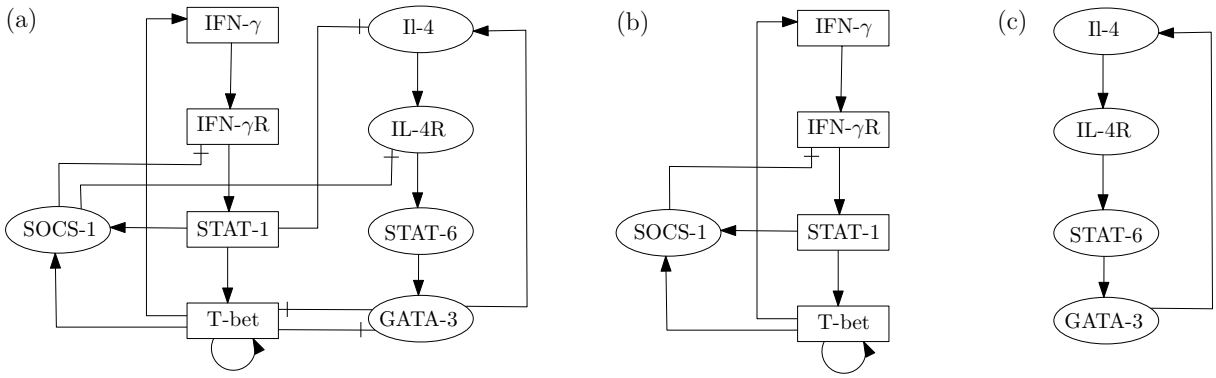



Figure 4: Subnetworks of the Th cell differentiation network associated with the symbolic fixed points derived from the input values $x_1 = x_2 = x_3 = 0$ in (a) and $x_1 = 1$, $x_2 = x_3 = 0$ in (b).

$x := (\theta, 0, \theta, 0, \theta, \theta, 0, \theta, 0)$ is a seed of $f^{G^\theta[s^2]}$. One iteration step shows that $x$ is also a symbolic steady state. The global interaction graph of the reduced system is shown in Fig. 4 (b). We derive three attractors of $f^{G^\theta[s^2]}$ from $x$, namely $(0,0,0,0,0,0,0,0,0)$, $(1,0,1,0,1,1,0,1,0)$ and $(2,0,1,0,1,1,0,2,0)$.

The circuit

$$(\alpha_5, \alpha_7, \alpha_{12}, \alpha_{16}) = (\text{IFN-}\gamma, \text{IFN-}\gamma\text{R}, \text{STAT-1}, \text{T-bet})$$

is also self-freezing. The assignment is given by the values $c_i := 0$, $i \in \{5, 7, 12, 16\}$, i. e., $(0, \theta, 0, \theta, \theta, 0, \theta, 0, \theta)$ is also a seed of $f^{G^\theta[s^2]}$. The corresponding symbolic steady state is $y := (0, \theta, 0, \theta, 0, 0, \theta, 0, \theta)$. The global interaction graph of the reduced system is shown in Fig. 4 (c). Here, we obtain two attractors of $f^{G^\theta[s^2]}$, namely $(0,0,0,0,0,0,0,0,0)$ and $(0,1,0,1,0,0,1,0,1)$.

Together, the two self-freezing circuits we considered here yield all attractors of the original system. However, the two seeds derived from them do not cover the state space of $f^{G^\theta[s^2]}$ in its entirety, which is the only criterion we have for ensuring that we obtain all attractors of the original system from the reduced systems (as mentioned in Sect. 6.1). Clearly, this is a point of interest for future work.

Thus far we have only focussed on reduction of complexity when analyzing the network using symbolic steady states. Also of interest is the identification of the subnetworks associated with the symbolic steady state, since they represent the characteristics of the system responsible for asymptotic dynamical behavior. For example, we have seen that input values $x_i = 0$, $i \in \{1, 2, 3\}$, completely determine the component values of IFN-$\beta$R, IL-2R, STAT4, IL-18R and IRAK in the attractors of the system. Their asymptotic behavior is independent of the behavior of the components belonging to the subnetwork represented by the graph in Fig. 4 (a). In contrast, the symbolic steady state $(1, 1, 1, 1, \theta, \ldots, \theta)$ derived from the input values $x_i = 1$, $i \in \{1, 2, 3\}$, has only four regular components. Therefore, the corresponding subnetwork consists of all components of the original system except for the three input vertices and the vertex IFN-$\beta$R. This is not very useful in terms of network reduction, however, we can derive information nonetheless. In this case, the system's asymptotic behavior heavily depends on components not belonging to the input network. In particular for signal transduction networks, where signal reception can easily be modeled using input networks, information this coarse is already

biologically relevant. A refined understanding of the relation between asymptotic behavior and subnetworks might then be possible using self-freezing circuits.

## 8 Conclusion and prospects

Analyzing complex networks is a difficult task. Even if the number of components of a discrete regulatory network is per se manageable, we have to deal with the problem of analyzing the dynamics in an exponentially large state space. To overcome this difficulty, a well-known approach is to identify smaller building blocks of the system the study of which in isolation still renders information on the dynamics of the whole network. In this paper, we introduce the notion of symbolic steady state which allows us to identify such building blocks and systematically extend ideas developed for Boolean functions and asynchronous dynamics in [34] and [32]. For synchronous as well as asynchronous dynamics, we state explicit rules on how to derive attractors of the network from subnetwork attractors. Illustrating these rules, we derive general conditions for circuits embedded in the network to transfer their behavioral characteristics pertaining to number and size of attractors observed in isolation to the complex network. We also propose methods for determining symbolic steady states based on structural and dynamical characteristics of the system.

Stronger results are possible if we refine the representation of component values via the symbolic value. Instead of merging the whole range of a component to one symbolic value we could partition it into several symbolic values, which would allow for a more precise localization in phase space. Such a refinement can be useful for analysis and poses no difficulty from a mathematical point of view, although careful consideration has to be given to the differences generated by the choice of update strategy. We presented some results for the asynchronous case in a workshop contribution [33], where we also considered the Th cell differentiation network. We have not yet considered refinements in the synchronous case.

An even more accurate understanding of the interactions governing asymptotic behavior in the region of state space associated with a symbolic steady state $x$ would be possible when considering the local interaction graph $G(M)$ for a set $M \subseteq [x]$ derived from $[x]$ by eliminating in some sense dynamically irrelevant states. For example, a comparison of $[x]$ with its forward orbit may be helpful.

To make the theory more accessible for testing and for using it in the study of logical models, we have to provide procedures for determining symbolic steady states that are suitable for implementation. We presented some ideas addressing this problem in Sect. 6. The results show that circuits in the interaction graph may be the key to a more comprehensive understanding. In particular, the relation between self-freezing circuits and functional circuits should be clarified. Here, the notion of functionality context as introduced in [18] could prove very useful.

Network analysis methods based on circuit functionality have been introduced and applied to a simplified Boolean T helper cell differentiation network in [23]. In contrast to the results presented here, those methods provide statements about the behavior of subsets of components that are not necessarily sufficient to derive the behavior of the network as a whole. The central ideas of both approaches seem to dovetail very nicely, but their relation is not obvious at first glance. Preliminary research shows that the approaches focus on different aspects of the link of structure and dynamics, thus further investigation might prove very fruitful.

A thorough understanding of the links between the different concepts may also allow for

easy integration of procedures calculating symbolic steady states into software capable of analyzing logical models with respect to circuits, as e. g. GINsim [16]. Circuits may also play an important role regarding the extent of network reduction possible using symbolic steady states. In general, networks that are not too densely connected may be better candidates, especially when determining symbolic steady states using seeds derived from input values. As mentioned in Sect. 6.1 we need at most as many iteration steps as there are symbolic components in the seed. However, the iteration procedure may terminate after only a few steps if we deal with densely connected networks. In this case, the information we gain is limited.

Future work will also focus on comparing our approach to other network modularization and reduction techniques. A good starting point would be a comparison with reduction techniques available for logical models (see e. g. [17]), but we should also consider methods proposed for other modeling frameworks to gain a clearer understanding of underlying concepts.

Lastly, a very important step is to apply the methods to established biological network models. This would allow not only for testing the suitability of the approach to the dynamical analysis, but also for a comparison of the subnetworks derived from symbolic steady states with network modules of known biological importance.

## References

[1] U. Alon. Network motifs: theory and experimental approaches. *Nat. Rev. Genet.*, 8:450–461, 2007.

[2] R. J. Bagley and L. Glass. Counting and classifying attractors in high dimensional dynamical systems. *J. Theoret. Biol.*, 183(3):269 – 284, 1996.

[3] U. Bastolla and G. Parisi. The modular structure of kauffman networks. *Phys. D*, 115(3-4):219 – 233, 1998.

[4] F. J. Bruggeman, H. V. Westerhoff, J. B. Hoek, and B. N. Kholodenko. Modular response analysis of cellular regulatory networks. *J. Theoret. Biol.*, 218(4):507 – 520, 2002.

[5] C. Christensen and R. Albert. Using graph concepts to understand the organization of complex systems. *Int. J. Bifurc. Chaos*, 17(7):2201–2214, 2007.

[6] E. Dubrova. Random multiple-valued networks: Theory and applications. In *Proceedings of 36th IEEE International Symposium on Multiple-Valued Logic (ISMVL'2006), Singapore*, pages 27–33, 2006.

[7] E. Dubrova and M. Teslenko. Compositional properties of random boolean networks. *Phys. Rev. E*, 71(5):056116, 2005.

[8] F. Harary, J. P. Hayes, and H.-J. Wu. A survey of the theory of hypercube graphs. *Comput. Math. Appl.*, 15(4):277 – 289, 1988.

[9] L. Hartwell, J. Hopfield, S. Leibler, and A. Murray. From molecular to modular cell biology. *Nature*, 402:C47 – C52, 1999.

[10] A. S. Jarrah, B. Raposa, and R. Laubenbacher. Nested canalyzing, unate cascade, and polynomial functions. *Phys. D*, 233(2):167 – 174, 2007.

[11] S. Kauffman. *The Origins of Order*. Oxford University Press, 1993.

[12] R. Laubenbacher and P. Mendes. A discrete approach to top-down modeling of biochemical networks. In *Computational Systems Biology*, pages 229–247. Elsevier Academic Press, 2005.

[13] L. Mendoza. A network model for the control of the differentiation process in Th cells. *Biosystems*, 84(2):101 – 114, 2006.

[14] L. Mendoza and E. R. Alvarez-Buylla. Dynamics of the genetic regulatory network for *Arabidopsis thaliana* flower morphognesis. *J. Theoret. Biol.*, 193(2):307 – 319, 1998.

[15] L. Mendoza, D. Thieffry, and E. R. Alvarez-Buylla. Genetic control of flower morphogenesis in arabidopsis thaliana: a logical analysis. *Bioinformatics*, 15:593 – 606, 1999.

[16] A. Naldi, D. Berenguier, A. Faur, F. Lopez, D. Thieffry, and C. Chaouiya. Logical modelling of regulatory networks with ginsim 2.3. *Biosystems*, 97(2):134 – 139, 2009.

[17] A. Naldi, E. Remy, D. Thieffry, and C. Chaouiya. A reduction of logical regulatory graphs preserving essential dynamical properties. In *Computational Methods in Systems Biology, CMSB 2009, Bologna, Italy*, volume 5688 of *LNBI*, pages 266–280. Springer, 2009.

[18] A. Naldi, D. Thieffry, and C. Chaouiya. Decision diagrams for the representation and analysis of logical models of genetic networks. In *Computational Methods in Systems Biology, CMSB 2007, Edinburgh, Scotland*, volume 4695 of *LNBI*, pages 233–247. Springer, 2007.

[19] M. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA*, 103(23):8577 – 8582, 2006.

[20] J. A. Papin, J. L. Reed, and B. O. Palsson. Hierarchical thinking in network biology: the unbiased modularization of biochemical networks. *Trends Biochem. Sci.*, 29(12):641 – 647, 2004.

[21] U. Paul, V. Kaufman, and B. Drossel. Properties of attractors of canalyzing random boolean networks. *Phys. Rev. E*, 73(2):026118, 2006.

[22] E. Remy, B. Mossé, C. Chaouiya, and D. Thieffry. A description of dynamical graphs associated to elementary regulatory circuits. *Bioinformatics*, 19:172–178, 2003.

[23] E. Remy and P. Ruet. From minimal signed circuits to the dynamics of Boolean regulatory networks. *Bioinformatics*, 24:220–226, 2008.

[24] E. Remy, P. Ruet, and D. Thieffry. Graphic requirements for multistability and attractive cycles in a boolean dynamical framework. *Adv. Appl. Math.*, 41(3):335–350, 2008.

[25] A. Richard. On the link between oscillations and negative circuits in discrete genetic regulatory networks. JOBIM, 10-12 juillet, Marseille, France, 2007.

[26] A. Richard. Positive circuits and maximal number of fixed points in discrete dynamical systems. *Discrete Appl. Math.*, 157(15):3281–3288, 2009.

[27] A. Richard and J.-P. Comet. Necessary conditions for multistationarity in discrete dynamical systems. *Discrete Appl. Math.*, 155(18):2403–2413, 2007.

[28] A. Richard, J.-P. Comet, and G.Bernot. R. Thomas' modeling of biological regulatory networks: introduction of singular states in the qualitative dynamics. *Fund. Inform.*, 65:373–392, 2005.

[29] F. Robert. *Discrete Iterations: A Metric Study*, volume 6 of *Springer Series in Computational Mathematics*. Springer, 1986.

[30] C. Robinson. *Dynamical Systems: Stability, Symbolic Dynamics, and Chaos*. CRC Press, 1998.

[31] P. Shetty, M.-C. Lo, S. M. Robertson, and R. Lin. C. elegans TCF protein, POP-1, converts from repressor to activator as a result of Wnt-induced lowering of nuclear levels. *Dev. Biol.*, 285(2):584 – 592, 2005.

[32] H. Siebert. Deriving behavior of boolean bioregulatory networks from subnetwork dynamics. *Math. Comput. Sci.*, 2(3):421–442, 2009.

[33] H. Siebert. Dynamical and structural modularity of discrete regulatory networks. In *Computational Models for Cell Processes, CompMod 2009, Eindhoven, Netherlands*, volume 6 of *EPTCS*, pages 109–124, 2009.

[34] H. Siebert and A. Bockmayr. Relating attractors and singular steady states in the logical analysis of bioregulatory networks. In *Algebraic Biology, AB 2007, Castle of Hagenberg, Austria*, volume 4545 of *LNCS*, pages 36–50. Springer, 2007.

[35] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. II. Logical analysis of regulatory networks in terms of feedback circuits. *Chaos*, 11:180–195, 2001.

[36] K. Willadsen and J. Wiles. Robustness and state-space structure of boolean gene regulatory models. *J. Theoret. Biol.*, 249(4):749 – 765, 2007.

# Chapter 5

# Dynamical and Structural Modularity of Discrete Regulatory Networks

Heike Siebert.

In Proceedings of the 2nd International Workshop on *Computational Models for Cell Processes*, CompMod 2009, Eindhoven, Netherlands, volume 6 of *EPTCS*, pages 109-124, 2009.

**Remarks.** In this paper, I further refine the framework introduced in the previous chapter, allowing for stronger results improving the application. The theoretical focus lies on a rigorous description of the method and the mathematical objects involved in the context of modularization.

# Dynamical and Structural Modularity of Discrete Regulatory Networks

Heike Siebert

*Abstract.* A biological regulatory network can be modeled as a discrete function $f$ that contains all available information on network component interactions. From $f$ we can derive a graph representation of the network structure as well as of the dynamics of the system. In this paper we introduce a method to identify modules of the network that allow us to construct the behavior of $f$ from the dynamics of the modules. Here, it proves useful to distinguish between dynamical and structural modules, and to define network modules combining aspects of both. As a key concept we establish the notion of symbolic steady state, which basically represents a set of states where the behavior of $f$ is in some sense predictable, and which gives rise to suitable network modules. We apply the method to a regulatory network involved in T helper cell differentiation.

## 1 Introduction

Qualitative methods present a rigorous mathematical framework for modeling biological systems for which experimental data needed to determine kinetic parameters and mechanisms is lacking. The components of the system are modeled as variables adopting only finitely many values, so-called activity levels. In the simplest case, we obtain a Boolean representation, where the values 0 and 1 may for example represent a gene being inactive or active. In the general case, each component can have several activity levels, which may be appropriate depending on the biological data, and often is useful when modeling components that influence several other network components. A vector assigning each component an activity level then represents a state of the system. The information about network structure as well as the logical rules governing the behavior of the system in state space is represented by a discrete function $f$.

Although discrete networks are a strongly simplified representation of the original system, complex networks are hard to analyze, not least because the state space grows exponentially with the number of components. So, methods to reduce the complexity of the analysis are of great interest. One approach is to deconstruct the network in smaller building blocks that can be analyzed more easily, which leads to the notion of network modularity.

The idea of decomposing networks into modules is well-established in systems biology, although the notion of network module is not clear-cut. Often modules are defined based on biological criteria, that have to be translated into mathematical properties in order to identify them in a mathematical model (see e. g. [5, 4, 12]). Other approaches focus purely on the the graph representation of the network structure. Modules are defined as subgraphs satisfying

graph theoretical characteristics often related to connectivity [3], or with statistical significance in comparison with random networks [11, 1]. In addition to this structural view, there are also approaches to find dynamical modules, see e. g. [6], that focus on identifying behavioral characteristics. However, in general the results obtained by analyzing such modules in isolation do not translate to the original network, since additional influences have to be taken into account once the module is re-embedded in the original system. Here, the key is finding conditions that allow to draw conclusions about a complex network from knowledge obtained from module analysis, as e. g. possible in the modular response analysis approach in the context of metabolic networks and steady state fluxes [9, 2].

In this paper, we focus on the discrete modeling approach, presenting a method to identify *network modules* that allow us to derive precise information on the dynamics of the original system from the results of the analysis of the modules, building on ideas and significantly extending results from [17, 16]. In particular, we show that we can explicitly construct attractors of the original systems from network module attractors. Here, modularity is a key concept, and we exploit a purely structural as well as a purely dynamical view of modularity to eventually determine network modules combining important aspects of both. The core notion in our method is that of *symbolic steady state*. Such a state represents a set of constraints on the activity levels of the network components that allows us on the one hand to focus on dynamics restricted to subsets of state space, on the other hand enables us to identify dynamical and structural modules that render the basis for defining suitable network modules.

The paper is organized as follows. In the next section we describe the discrete modeling formalism we use throughout the paper, and introduce structural, dynamical and network modules. In Sect. 3 we establish the notion of symbolic steady state as well as related concepts. This is followed by the main results concerning network analysis utilizing modules in Sect. 4. We then illustrate the results for a class of networks, namely networks with input layer. In Sect. 6 we apply the method to the analysis of a regulator network involved in T helper cell differentiation proposed in [10]. We close with conclusions and perspectives.

## 2 Discrete regulatory networks

In this paper we model regulatory systems as discrete functions which capture all available information about network interactions and the logical rules governing the behavior of the system. Throughout the text, let us consider a system consisting of $n \in \mathbb{N}$ network components $\alpha_1, \ldots, \alpha_n$. In the following we identify a component $\alpha_i$ with its index $i$ to simplify notation. Each component is interpreted as a variable which takes integer values that represent the different activity levels of the component. The literal meaning of those levels may be very different for different network components, for example they can represent levels of substance concentration, gene activity, presence or absence of a signal and so on. A vector assigning each component an activity level represents a state of the system, and the dynamics of the system is represented by state changes due to component interactions.

**Definition 2.1.** *For all $i \in \{1, \ldots, n\}$, let $p_i \in \mathbb{N}$, and set $X_i = \{0, 1, \ldots, p_i\}$. Set $X = X_1 \times \cdots \times X_n$, and let $f = (f_1, \ldots, f_n) : X \to X$ be a function. We call $f$ a* network *comprising $n$ components. For each $i \in \{1, \ldots, n\}$, the value $p_i$ is the* maximal activity level *of $\alpha_i$, and $X_i$ is called the* range *of $\alpha_i$. The set $X$ is called the* state space *of $f$.*

Each coordinate function $f_i$ of $f$ describes the rules governing the behavior of the $i$-th network component depending on the state of the system. But $f$ carries not only dynamical but also structural information on the system. Both aspects can be represented by directed graphs derived from $f$ as we will see in the following two subsections. In the remainder of the paper $f$ denotes a network as introduced in Def. 2.1

## 2.1 Structure

We represent the structure of a network by a signed directed (multi-)graph, where vertices represent the network components, and an edge from $\alpha_i$ to $\alpha_j$ signifies that the value of $f_j$ depends on the activity level of $\alpha_i$. The sign of the edge represents the character, i. e., activating or inhibiting, of the interaction. This description is inherently local in nature, so we first introduce a structural representation depending on the state of the system. This notion was introduced for Boolean functions in [13] and is used for multi-value functions in the form considered here in [14].

**Definition 2.2.** *Let $x \in X$. By $G(f)(x)$ we denote the directed signed (multi-)graph with vertex set $V = \{\alpha_1, \ldots, \alpha_n\}$ and edge set $E(x) \subseteq V \times V \times \{+, -\}$. An edge $(i, j, \varepsilon)$ belongs to $E(x)$ iff there exists $c_i \in \{-1, +1\}$ such that $x_i + c_i \in X_i$ and*

$$\operatorname{sgn}\left(\frac{f_j((x_1, \ldots, x_{i-1}, x_i + c_i, x_{i+1}, \ldots, x_n)) - f_j(x)}{c_i}\right) = \varepsilon\,,$$

*where the function $\operatorname{sgn} : \mathbb{Z} \to \{+, -, 0\}$ satisfies $\operatorname{sgn}(0) = 0$, $\operatorname{sgn}(z) = +$ if $z > 0$, and $\operatorname{sgn}(z) = -$ if $z < 0$. We call $G(f)(x)$ the* local interaction graph of $f$ in $x$.

The local interaction graph in $x$ is closely related to the discrete Jacobian matrix as introduced in [15] in the Boolean case. Note that in the multi-value other than in the Boolean case it is possible that $G(f)(x)$ contains parallel edges. There are at most two parallel edges from one vertex to another which then have opposite sign.

The local definition is easily extended, if we are interested in a representation of the interactions influencing the system behavior in larger subsets of state space.

**Definition 2.3.** *Let $Y \subseteq X$. We denote by $G(f)(Y)$ the union of the graphs $G(f)(x)$, $x \in Y$. We denote the graph $G(f)(X)$ also by $G(f)$ and call it the* global interaction graph of $f$.

In Fig. 1 (b) we see the global interaction graph of the network defined in Fig. 1 (a). The local interaction graph in the state $(1, 1, 0)$ is shown in Fig. 2 (a). To simplify notation we often write $G(x)$ and $G(Y)$ instead of $G(f)(x)$ and $G(f)(Y)$, respectively, if the corresponding function $f$ is clear from the context.

When analyzing interaction graphs we are in particular interested in modules of the graph, a term for which there exists a variety of definitions as mentioned in the introduction. For our purposes it is convenient to use the term in the broadest sense, initially.

**Definition 2.4.** *A directed (multi-)graph $G' = (V^{G'}, E^{G'})$ is called a subgraph of a directed (multi-)graph $G = (V^G, E^G)$ if $V^{G'} \subseteq V^G$, $E^{G'} \subseteq E^G$, and every edge in $E^{G'}$ has both end-vertices in $V^{G'}$. We call a subgraph of $G(f)$* structural module of $f$.

(a)

$f : \{0,1,2\}\times\{0,1\}\times\{0,1\} \to \{0,1,2\}\times\{0,1\}\times\{0,1\}$

$$
\begin{aligned}
f_1(x) &= 0, \text{ if } x_1 = x_2 = 0 \wedge x_3 = 1,\\
f_1(x) &= 2, \text{ if } x_1 = 2 \wedge x_2 = 1,\\
f_1(x) &= 1, \text{ otherwise,}\\
\\
f_2(x) &= 1, \text{ if } (x_1 \geq 1) \vee (x_2 = 1),\\
f_2(x) &= 0, \text{ otherwise,}\\
\\
f_3(x) &= x_2 \wedge \neg\, x_3
\end{aligned}
$$

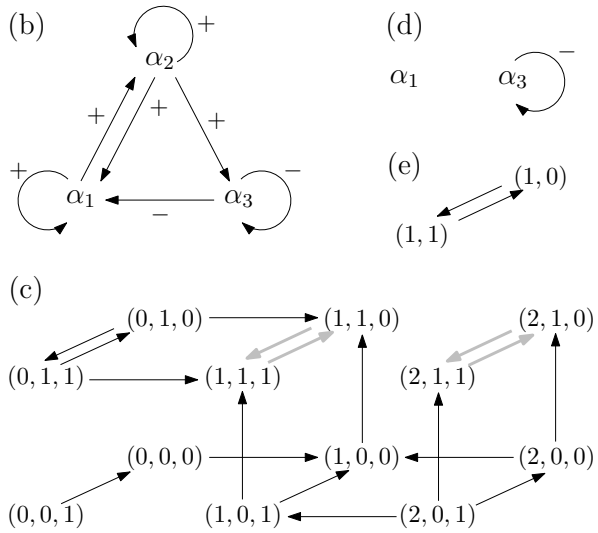



Figure 1: A network (a), its global interaction graph (b), and its state transition graph (c), where attractors are indicated by fat gray edges. Logical disjunction, conjunction and negation are represented by $\vee$, $\wedge$, and $\neg$, respectively. In (d) a structural, in (e) a dynamical module of $f$.

Note that for a structural module the vertex set, the edge set or both may be smaller than for $G(f)$. For example, the graph shown in Fig. 1 (d) is a structural module of the function $f$ given in the same figure. In general, local interaction graphs are structural modules of $f$. Of course, this definition is not very useful for analyzing the network structure or finding characteristics of the system. However, it is of conceptual advantage for us in the endeavor of defining network modules that combine structural and dynamical characteristics.

## 2.2 Dynamics

There are different approaches to deriving the dynamics of $f$. Commonly used is the so-called synchronous update strategy, where the successor of a state $x$ is its image under $f$. A more realistic assumption is that not all changes indicated by differences in component values of $x$ and $f(x)$ take the same amount of time to be executed, since they may represent very different biological processes. However, we lack the information to decide which of those processes of activity level change is the fastest. Therefore, all possible state transitions are taken into account leading to a non-deterministic representation of the dynamical behavior. Furthermore, we assume that a component value changes only by absolute value one in each transition, even though the function value may indicate a bigger change. This update method is called *asynchronous update* [18, 19].

**Definition 2.5.** *We denote by $S(f)$ the directed graph with vertex set $X$ and edge set $E(S(f))$ defined as follows. We have $(x, x') \in E(S(f))$ for states $x = (x_1, \ldots, x_n)$, $x' = (x'_1, \ldots, x'_n) \in X$ if and only if $x' = f(x) = x$, or $x'_i = x_i + \mathrm{sgn}(f_i(x) - x_i)$ for some $i \in \{1, \ldots, n\}$ with $x_i \neq f_i(x)$ and $x'_j = x_j$ for all $j \neq i$. We call $S(f)$ the* asynchronous state transition graph of $f$.

To analyze state transition graphs we use, in addition to standard terminology from graph theory such as paths and cycles, the following concepts.

**Definition 2.6.** *An infinite path $(x^0, x^1, \dots)$ in $S(f)$ is called* trajectory. *A nonempty set of states $D$ is called* trap set *if every trajectory starting in $D$ never leaves $D$. A trap set $A$ is called* attractor *if for all $x^1, x^2 \in A$ there is a path from $x^1$ to $x^2$ in $S(f)$. Attractors of cardinality greater than one are called* cyclic *attractors. A state $x$ is called* steady state, *if there exists an edge $x \to x$, i. e. if $f(x) = x$.*

It is easy to see that each trap set contains at least one attractor, and that attractors are the terminal strongly connected components of $S(f)$. They represent asymptotically stable behavior and often have clear biological meaning.

In Fig. 1 (c) we see the state transition graph for the network introduced in (a). The system has two cyclic attractors, namely $\{(1,1,0),(1,1,1)\}$ and $\{(2,1,0),(2,1,1)\}$.

As for the structural graph, we can define modules of the state transition graph as subgraphs, i. e. sets of states and corresponding state transitions representing fractions of the system's dynamics. However, it may also be of interest to only focus on the behavior of a subset of network components, which we can derive from the state transition graph by projection.

**Definition 2.7.** *Let $S' = (Y, E(S'))$ be a subgraph of $S(f)$. Let $\pi^I : X \to \prod_{i\in I} X_i$ be the projection on the components in the ordered set $I \subseteq \{1,\dots,n\}$. We define $\pi^I(S')$ as the graph with vertex set $\pi^I(Y)$ and edges $\pi^I(v^1) \to \pi^I(v^2)$ for $v^1, v^2 \in Y$ such that there exists an edge $v^1 \to v^2$ in $S(f)$, and $v^1 = v^2$ or $v_i^2 = v_i^1 + \operatorname{sgn}(f_i(v^1) - v_i^1)$ for some $i \in I$. We call $\pi^I(S')$* dynamical module of $f$.

Any subgraph of $S(f)$ is a dynamical module of $f$, with $I$ in the above definition chosen as the set $\{1,\dots,n\}$. Fig. 1 (e) shows a dynamical module of the function $f$ given in the same figure. Here, we choose the subgraph $S'$ consisting of the cyclic attractor $\{(1,1,0),(1,1,1)\}$ and the corresponding edges between the two attractor states. Then the dynamical module $\pi^{\{1,3\}}(S')$ of $f$ is the graph in Fig. 1 (e).

Again, we do not incorporate any restrictions in the definition that ensure a significance of the modules, as e. g. in the approach in [6] where the authors focus on projected dynamics that are independent of the behavior of the rest of the system.

## 2.3 Network modules

As mentioned in the introduction, we are looking for subnetworks of $f$ that are on the one hand easier to analyze than $f$ itself, and on the other hand carry information of importance for understanding the original system. We define modules of the network $f$ as follows.

**Definition 2.8.** *We call a function $g$ a* network module of $f$, *if there exist $Y \subseteq X$ and an ordered set $I \subseteq \{1,\dots,n\}$ such that $g : \pi^I(Y) \to \pi^I(X)$ satisfies*

$$\pi^I \circ f|_Y = g \circ \pi^I|_Y,$$

*where $|_Y$ denotes the restriction of a function to the set $Y$.*
*We call $g$* autonomous, *if there exist integer intervals $Z_i = \{a_i, a_i + 1, \dots, b_i\}$, $a_i \le b_i$, for all $i \in \{1,\dots,k\}$, $k = \operatorname{card} I$, such that $\pi^I(Y) = Z_1 \times \cdots \times Z_k$, and if $g(\pi^I(Y)) \subseteq \pi^I(Y)$.*

Let us again illustrate the notion using the example introduced in Fig. 1.
For $Y = \{(1,1,0),(1,1,1)\}$, we have $f(Y) = \{(1,1,0),(1,1,1)\}$. If we set $I = \{1,3\}$ and

$$g : \{1\} \times \{0,1\} \to \{0,1,2\} \times \{0,1\}, \quad g((1,1)) = (1,0) \text{ and } g((1,0)) = (1,1)\,,$$

then $\pi^I(f(x)) = g(\pi^I(x))$ for $x \in Y$. Since $g(\pi^I(Y)) \subseteq \pi^I(Y)$ holds, $g$ is an autonomous network module of $f$. Note that the set $Y$ is a set on which the behavior of $f$ can in some sense be characterized by the behavior of the components in $I$. If we add e. g. the state $(1,0,0)$ to the set $Y$ in our example, then there is no function $g$ satisfying $\pi^I \circ f|_Y = g \circ \pi^I|_Y$, since $\pi^I(1,0,0) = \pi^I(1,1,0)$, but $\pi^I(f(1,0,0)) = (1,0) \neq (1,1) = \pi^I(f(1,1,0))$. That is, we cannot distinguish the behavior of $f$ in states $(1,0,0)$ and $(1,1,0)$ if we only have information on the components in $I$.

In general, network modules represent rather local aspects of the network in the sense that they describe the influences acting on a subset of components in a set of states. However, the information inherent in a network module does not necessarily suffice for determining dynamics beyond a single transition step. The second condition for autonomous network modules $g$ allows to apply $g$ iteratively on states in $\pi^I(Y)$, while the first ensures that we can derive trajectories according to the asynchronous update rule in a projection of state space. Moreover, the first condition allows to apply Def. 2.2 to $g$. Thus, for autonomous network modules $g$ we can determine an interaction graph $G(g)$ and a state transition graph $S(g)$. By abuse of notation we also denote $G(g)$ the graph derived from $G(g)$ by renaming the vertices $1,\ldots,k$ of $G(g)$ with the indices in $I$ while preserving the order. This allows us to identify the interaction graph of $g$ with a subgraph of $G(f)$.

**Lemma 2.9.** *Let $g$ be an autonomous network module as introduced in Def. 2.8. Then $G(g)$ is a structural and $S(g)$ is a dynamical module of $f$.*

*Proof.* As already mentioned, we associate each $i \in I$ with $l^i \in \{1,\ldots,k\}$ via an order-preserving mapping, and rename each vertex of $G(g)$ with indices in $I$ according to this mapping. Obviously, the vertex set of $G(g)$ resp. $S(g)$ is a subset resp. a projection via $\pi^I$ of a subset of the vertex sets of $G(f)$ resp. $S(f)$. Let $i \in I$, and choose $l^i \in \{1,\ldots,k\}$ as above, i. e., $\pi^i$ maps the $i$-th component of a state $x \in X$ to the $l^i$-th component in $\pi^I(X)$. Then we have $f_i(y) = (\pi^I(f(y)))_{l^i} = g_{l^i}(\pi^I(y))$ for all $y \in Y$. Application of this equation to the conditions defining edges in Def. 2.2 and 2.5 easily renders that each edge in $G(g)$ is also an edge of $G(f)$, and that $S(g) = \pi^I(S')$, where $S'$ denotes the subgraph of $S(f)$ with vertex set $Y$ and edges $y^1 \to y^2$ of $S(f)$ with $y^1, y^2 \in Y$. □

For the network module $g$ as defined as illustration for Def. 2.8 the state transition graph is shown in Fig. 1 (e). We rename the vertex set $\{1,2\}$ of $G(g)$ with $I = \{1,3\}$. The graph $G(g)$ then consists of the vertices $\alpha_1$ and $\alpha_3$ and a negative loop on $\alpha_3$ as shown in Fig. 1 (d).

In the following sections, we focus on developing a method to determine network modules useful in the analysis of $f$.

## 3 Symbolic steady states and frozen components

Often network components are involved in a number of specific tasks. Thus, although a network component may have a large range, only subsets of the range may be of interest when focusing

on specific network behavior. To exploit this observation, we introduce the following notation. Here, we call the set $[a_i, b_i] := \{a_i, a_i + 1, \ldots, b_i - 1, b_i\} \subseteq X_i$ a discrete interval, if $a_i \leq b_i$, with $[a_i, a_i] := \{a_i\}$. In the following, we identify $\{a_i\}$ with $a_i$ for all $a_i \in X_i$, $i \in \{1, \ldots, n\}$, and call $a_i$ *regular value*. We will use intervals of cardinality greater than one instead of regular component values, if we do not have enough information to determine the exact component value. Following the terminology in [17, 16], we call intervals $[a_i, b_i]$ with $a_i < b_i$ *symbolic values*.

We now need to integrate symbolic values in the dynamical analysis. Here, we generalize ideas from [17, 16].

**Definition 3.1.** *For every $i \in \{1, \ldots, n\}$ let*

$$X_i^{\square} := \{[a_i, b_i] \subseteq X_i \mid a_i \leq b_i\}$$

*of discrete intervals in the range $X_i$. Set $X^{\square} = X_1^{\square} \times \cdots \times X_n^{\square}$. We call elements in $X$* regular*, elements in $X^{\square} \setminus X$* symbolic states*. By $J(M)$ we denote the set of all symbolic valued components of $M$ for $M \in X^{\square}$. Define*

$$F : X^{\square} \to X^{\square},\ M \mapsto (F_1(M), \ldots, F_n(M)) \ \text{ with } \ F_i(M) = [\min_{x \in M} f_i(x), \max_{x \in M} f_i(x)]$$

*for all $i \in \{1, \ldots, n\}$. We call a state $M \in X^{\square} \setminus X$ satisfying $F(M) = M$* symbolic steady state*.*

The elements of $X^{\square}$ are subsets of $X$. The functions $f$ and $F$ coincide on the set $X$ of regular states which we identify with the elements of $X^{\square}$ of cardinality one. In general, if a component function value $f_i(M)$ is regular, this means there is enough information inherent in $M$ to exactly specify its value, while a symbolic value represents the fact that we have not enough information to do so. However, it may be possible to at least derive some constraint for the function value represented by the interval boundaries. For our running example given in Fig. 1 the state $([1,2], 1, [0,1]) = ([1,2], 1, X_3) = F(([1,2], 1, X_3))$ is a symbolic steady state, where we can determine $F_2(([1,2], 1, [0,1])) = 1$ exactly, obtain the constraint that the first component cannot have value 0, but have no information on the third component.

We are particularly interested in regular components of a symbolic state $M$ that remain fixed on all trajectories starting in $M$. Keeping in mind that we consider the asynchronous update strategy, we can find a superset of the set of states reachable from $M$ by the following procedure. We define $\tilde{M}^0 := M$ and $\tilde{M}_j^k := [\min(\tilde{M}_j^{k-1} \cup F_j(\tilde{M}^{k-1})), \max(\tilde{M}_j^{k-1} \cup F_j(\tilde{M}^{k-1}))]$ for all $k \in \mathbb{N}$. Since the boundaries of the intervals $M_j^k$ decrease resp. increase monotonously and are bounded by 0 resp. the maximal activity level $p_j$, the sequence $(\tilde{M}^k)_{k \in \mathbb{N}_0}$ converges to a symbolic state $\widetilde{M}$ representing a superset of the set of from $M$ in $S(f)$ reachable states. In particular, no trajectory starting in $\widetilde{M}$ can leave $\widetilde{M}$. We call $\widetilde{M}$ *extended forward orbit of $M$*. The next definition is in reference to the notion of frozen cores in random Boolean networks introduced by S. Kauffman [8].

**Definition 3.2.** *Let $i \in \{1, \ldots, n\}$. If $M$ is a symbolic state with regular component $M_i$ such that $\widetilde{M}_i = M_i$ for the extended forward orbit $\widetilde{M}$ of $M$, then we say that the $i$-th network component is a* frozen component of $M$*, or a component* frozen to value $M_i$*. The set $I$ of all frozen components of a symbolic state $M$ is called* frozen core *of $M$, and is denoted by $(I, M)$.*

If a component $j$ of a symbolic state $M$ has symbolic value $X_j$, then of course the $j$-th component of the extended forward orbit is also $X_j$. For the example network in Fig. 1 the symbolic state $M = (X_1, 1, X_3)$ coincides with its extended forward orbit. Thus, the frozen core of $M$ is given by $(\{2\}, (X_1, 1, X_3))$.

Clearly, the frozen core of a symbolic steady state coincides with its set of regular components. Moreover, we can use the frozen core of a symbolic state to obtain a symbolic steady state, as the next statement shows.

**Theorem 3.3.** *Let $(I, M')$ be the frozen core of a symbolic state $M' \in X^{\square}$. Set $M^0 := \widetilde{M'}$ and $M^k := F(M^{k-1})$ for all $k \in \mathbb{N}$. Then $(M^k)_{k\in\mathbb{N}}$ converges to a regular or a symbolic steady state $M$. We call $M$ the (symbolic) steady state derived from $(I, M')$.*

*Proof.* If the sequence converges to a limit $M$, clearly $F(M) = M$ follows from the definition of the sequence. The state $M$ is a regular or a symbolic steady state depending on the cardinality of $M$. We show convergence of $(M^k)_{k\in\mathbb{N}}$ by proving via induction that the sequence is decreasing monotonously with respect to the subset relation. That is, we show $M^{l+1} \subseteq M^l$ for all $l \in \mathbb{N}$.

For $i \in I$, we have $M_i^1 = M_i^0$ by the definition of frozen components. If $i \in J(M^0) = \{1, \dots, n\} \setminus I$, we have $M_i^1 \subseteq M_i^0$ by the definition of the extended forward orbit.

Now, let $l \in \mathbb{N}$ and assume $M^k \subseteq M^{k-1}$ for all $k \leq l$. Recall that

$$M_i^{l+1} = F(M^l)_i = [\min_{x\in M^l} f_i(x), \max_{x\in M^l} f_i(x)] \qquad M_i^l = F(M^{l-1})_i = [\min_{x\in M^{l-1}} f_i(x), \max_{x\in M^{l-1}} f_i(x)]$$

for all $i \in \{1, \dots, n\}$. Since $M^l \subseteq M^{l-1}$, we have

$$\min_{x\in M^{l-1}} f_i(x) \leq \min_{x\in M^l} f_i(x) \leq \max_{x\in M^l} f_i(x) \leq \max_{x\in M^{l-1}} f_i(x)\,,$$

and thus $M_i^{l+1} \subseteq M_i^l$ for all $i \in \{1, \dots, n\}$. □

As mentioned above, for our running example $(\{2\}, (X_1, 1, X_3))$ is the frozen core of the state $(X_1, 1, X_3)$, where $X_1 = \{0, 1, 2\}$ and $X_3 = \{0, 1\}$. Since the state coincides with its extended forward orbit, we start the sequence with $M^0 = (X_1, 1, X_3) = ([0, 2], 1, [0, 1])$, and $M^1 = ([1, 2], 1, [0, 1]) = M$ is the symbolic steady state derived from $(\{2\}, (X_1, 1, X_3))$.

## 4 Network analysis using modules

In the following we want to determine network modules, such that the results of the module analysis can be directly used to obtain information on the behavior of the original system. Clearly, identification of such modules is generally only possible if we exploit at least some coarse knowledge of structural as well as dynamical characteristics of the original system. It turns out that the information inherent in a symbolic steady state $M$ is sufficient to determine network modules. We proceed by first associating a dynamical module with $M$, then derive a structural module, and finally define a network module suitable for utilization in network analysis.

Let us start by analyzing dynamical characteristics associated with a symbolic steady state $M$. If $x \in M$ is a regular state, then $f(x) \in M$ by definition of $F$. To be more precise, $x_i + \operatorname{sgn}(f_i(x) - x_i) \in M_i$, since the interval bounded by $x_i$ and $f_i(x)$ is a subset of $M_i$. Thus, every trajectory starting in $M$ remains in $M$. We formulate this fact in the following statement.

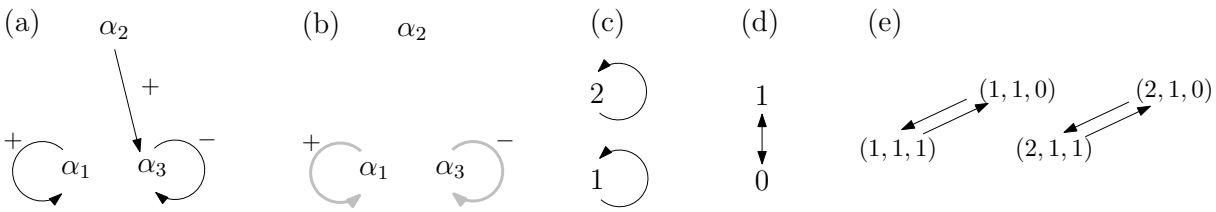



Figure 2: Consider $f$ as given in Fig. 1. In (a) the local interaction graph $G((1,1,0))$ of $f$. In (b) the graph $G(f|_M)$ for the symbolic steady state $M = ([1,2],1,[1,0])$. Here, heavier gray edges indicate the two components of $G^\theta(M)$. In (c), (d) and (e) the state transition graphs $S(f^{Z_1})$, $S(f^{Z_2})$ and $S^M$, respectively.

**Proposition 4.1.** *If $M$ is a symbolic steady state, then the set of regular states represented by $M$ is a trap set.*

By definition, the subgraph of $S(f)$ with vertex set $M$ and all edges in $S(f)$ with both end-vertices in $M$ is a dynamical module. The result above shows that this module is of significance for the dynamical analysis of $f$, since a trap set always contains at least one attractor.

In order to associate a structural module with $M$, we have to recognize some important properties attached to the regular components of $M$. While symbolic components may be dynamically active in the trap set $M$, i. e., the components change their values along at least some trajectories in $M$, the regular components remain fixed regardless of the behavior of the symbolic components in $M$. This means that the network components with symbolic values, i. e., components in $J(M)$, do not influence the behavior of the network components with regular values in the trap set $M$. In turn, the influence of the regular on the symbolic components remains the same for all states in $M$. This motivates the following definition describing the structural modules associated with $M$.

**Definition 4.2.** *Let $M$ be a symbolic steady state. By $G^\theta(M)$ we denote the (multi-)graph with vertex set $V^\theta(M) = J(M)$ and edge set $E^\theta(M) = \{(i,j,\varepsilon) \in E^M \mid i,j \in J(M)\}$, where $E^M$ denotes the edge set of $G(f|_M)$.*

*We call a graph $Z = (V^Z, E^Z)$* component *of $G^\theta(M)$, if the undirected graph derived from $Z$ is a maximal connected subgraph of the undirected graph derived from $G^\theta(M)$.*

Note that we use the global interaction graph $G(f|_M)$ instead of the local interaction graph $G(M)$ in the definition. The difference is that we only consider edges derived from component value changes in $M$ instead of in $X$ (compare Def. 2.2 and 2.3), and thus capture all interactions functional in $M$. In particular, there are no edges originating in frozen components of $M$, since it is not possible to vary their value without leaving $M$.

To illustrate the above notions let us again consider the example introduced in Fig. 1. We have seen that the state $M = ([1,2],1,[1,0])$ is a symbolic steady state. In Fig. 1 (c) we can easily see that the set of regular states $x \in M$ is a trap set which contains both attractors of the system. The global interaction graph $G(f|_M)$ is shown in Fig. 2 (b). We obtain $G^\theta(M)$ simply by eliminating the vertex $\alpha_2$. The two components of $G^\theta(M)$ are the loops originating in $\alpha_1$ resp. $\alpha_3$.

The components of $G^\theta(M)$ are the structural modules we associate with $M$. In preparation for our definition of a network module derived from a structural module we need to verify that

network components belonging to different components of $G^\theta(M)$ do not influence each others behavior in $M$. This property is captured in the following lemma, which has already been proved under slightly different conditions in [17, 16].

**Lemma 4.3.** *Let $M$ be a symbolic steady state, and let $Z_1, \dots, Z_k$ be the components of $G^\theta(M)$. Consider a union $Z$ of arbitrary components $Z_j$. Let $x, y \in M$ such that $x_i = y_i$ for all $i \notin Z$. Then $f_i(x) = f_i(y)$ for all $i \notin Z$.*

*In particular, for $M' \in X^\square$ such that $M'_i = M_i$ for all $i \notin Z$ and $M'_i \subseteq M_i$ for $i \in Z$, we have $F_i(M') = F_i(M) = M_i = M'_i$ for all $i \notin Z$.*

*Proof.* For $i \notin J(M)$, i. e., for a frozen component $i$ of $M$, we have $f_i(x) = f_i(y) = F_i(M)$ since $x, y \in M$.

Let $i \in J(M) \setminus Z$, and assume $f_i(x) \neq f_i(y)$. We know that if $x_j \neq y_j$ then $j \in Z$. Since $x, y \in M$, we can define a sequence $(x = x^1, x^2, \dots, x^m = y)$ in $M$ such that $x^l$ and $x^{l+1}$ differ in one component only, which is in $Z$, and the corresponding component values differ by absolute value one. Since $f_i(x) \neq f_i(y)$, it follows that $f_i(x^l) \neq f_i(x^{l+1})$ for some $l < m$. According to Def. 2.2 there exists an edge in $G(f|_M)$ from some component in $Z$ to $i$, which is a contradiction. □

The lemma shows that the frozen core of $M$ constitutes a boundary between the components of $G^\theta(M)$ that enables us to analyze their behavior in isolation from each other. To do so, we now derive a network module from a component $Z$ of $G^\theta(M)$ by defining a function $f^Z$.

**Lemma 4.4.** *Let $M$ be a symbolic steady state, and let $Z = (V^Z, E^Z)$ be a component of $G^\theta(M)$. Set $k^Z = \operatorname{card} V^Z$. Let $\iota^Z$ be an order preserving bijection from $\{1, \dots, k\}$ to $V^Z$. Set $X^Z = M_{\iota^Z(1)} \times \cdots \times M_{\iota^Z(k)}$. Let*

$$f^Z : X^Z \to X^Z\,, \quad f^Z = \pi^Z \circ F \circ \rho^Z\,,$$

*where $\rho^Z : X^Z \to X^\square$ with $\rho^Z_i(z) = M_i$ for $i \notin Z$ and $\rho^Z_i(z) = z_{\iota^Z(i)}$ for $i \in Z$, and the function $\pi^Z : X^\square \to X^Z$ is the projection on the components of $Z$. Then $f^Z$ is an autonomous network module, and is called the* network module of $f$ derived from $Z$.

*Proof.* The function $f^Z$ maps regular states to regular states, since $Z$ is disjoint from $J(M) \setminus Z$ in $G^\theta(M)$ and thus $F_i(\rho^Z(z)) \in X_i$ for $z \in X^Z$, $i \in Z$, according to Lemma 4.3 and the definition of $F$. We now show

$$\pi^Z \circ f|_M = f^Z \circ \pi^Z|_M\,.$$

Let $j \in \{1, \dots, k\}$ and $x \in M$. To simplify notation we drop the superscript $Z$ from $\pi^Z$, $\rho^Z$ and $\iota^Z$. We have $F_{\iota(j)}(\rho(\pi(x))) = F_{\iota(j)}(x) = f_{\iota(j)}(x)$ according to Lemma 4.3 and the definition of $\rho$ and $F$. Thus,

$$(\pi^Z \circ f|_M(x))_j = f_{\iota(j)}(x) = F_{\iota(j)}(x) = F_{\iota(j)}(\rho(\pi(x))) = (\pi \circ F \circ \rho(\pi(x)))_j = f^Z_j(\pi(x))\,.$$

Furthermore, since $M$ is a symbolic steady state $X^Z = \pi(M)$ satisfies the conditions regarding the domain of an autonomous network module given in Def. 2.8. □

It is easy to see that the global interaction graph $G(f^Z)$ is isomorphic to $Z$, so the structural module derived from $M$ matches the one derived from $f^Z$. The dynamical modules derived from

all functions $f^Z$, $Z$ component of $G^\theta(M)$, i. e., the state transition graphs $S(f^Z)$, constitute a breakdown of the coarse dynamical module, i. e., the trap set $M$, which we used to determine first the structural and then the network modules. We end this section by showing that these finer dynamical modules are building blocks of the dynamics of $f$. More specific, we show that we can compose the state transition graph of $f$ from the state transition graphs of the network modules $f^Z$ and the frozen components of $M$. Again, in the following we generalize results from [17]. First, we define the composition of the graphs $S(f^Z)$.

**Definition 4.5.** *Let $M \in X^\square$ be a symbolic steady state, and let $Z_1, \ldots, Z_m$ be the components of $G^\theta(M)$. We then denote by $S^M$ the graph with vertex set $M$ and edge set $E^M$. An edge $x^1 \to x^2$ belongs to the edge set iff*

$$x^1 = x^2, \quad \textit{and} \quad \pi^{Z_j}(x^1) \to \pi^{Z_j}(x^2) \quad \textit{belongs to } S(f^{Z_j}) \textit{ for all } j \in \{1, \ldots, m\},$$

*or if there exists $j \in \{1, \ldots, m\}$ such that*

$$\pi^{Z_j}(x^1) \to \pi^{Z_j}(x^2) \textit{ is an edge in } S(f^{Z_j}) \textit{ and } x^1_i = x^2_i \textit{ for all } i \notin V^{Z_j}.$$

*We call $S^M$ the product state transition graph corresponding to $M$.*

The next theorem confirms that the method of composing the state transition graphs of the network modules renders the subgraph of $S(f)$ derived from the state set $M$.

**Theorem 4.6.** *Let $M \in X^\square$ be a symbolic steady state, and let $Z_1, \ldots, Z_m$ be the components of $G^\theta(M)$. Let $S(f)|_M$ denote the subgraph of $S(f)$ with vertex set $M$ and all edges in $S(f)$ with both end-vertices in $M$. Then $S(f)|_M = S^M$.*

*Proof.* Recall that $M$ is a trap set, and that for $x, x' \in M$ there is an edge $x \to x'$ in $S(f)$ if and only if $x' = f(x) = x$ or $x'_i = x_i + \operatorname{sgn}(f_i(x) - x_i)$ for some $i \in \{1, \ldots, n\}$ satisfying $x_i \neq f_i(x)$, and $x'_j = x_j$ for all $j \neq i$.

First, we note that $f_i(x) = x_i = M_i$ for all $x \in M$ and $i \notin J(M)$. Furthermore, for $i \in J(M)$ there exists $k^i \in \{1, \ldots, m\}$ such that $i \in V^{Z_{k^i}}$, and there exists $l^i$ such that $\iota^{Z_{k^i}}(l^i) = i$, with $\iota^{Z_{k^i}}$ being the bijection introduced in Def. 4.4. As seen in the proof of Lemma 2.9, we have

$$f_i(x) = f^{Z_{k^i}}_{l^i}(\pi^{Z_{k^i}}(x))\,.$$

Therefore, $x \in M$ is a fixed point of $f$ iff $\pi^{Z_j}(x)$ is a fixed point of $f^{Z_j}$ for all $j \in \{1, \ldots, m\}$, and

$$x_i + \operatorname{sgn}(f_i(x) - x_i) = (\pi^{Z_{k^i}}(x))_{l^i} + \operatorname{sgn}(f^{Z_{k^i}}_{l^i}(\pi^{Z_{k^i}}(x)) - (\pi^{Z_{k^i}}(x))_{l^i})\,.$$

Thus, we can construct each edge in $S(f)|_M$ from an edge in some $S(f^{Z_j})$ and vice versa. It follows from Def. 4.5 that $S(f)|_M = S^M$. □

The reasoning in the proof of Theorem 4.6 leads immediately to the following statement.

**Corollary 4.7.** *Let $M$ and $Z_1, \ldots, Z_m$ be as in Theorem 4.6. For all $i \in \{1, \ldots, m\}$ let $A_i$ be an attractor in $S(f^{Z_i})$. Then $A := \{a \in M \mid \forall j \in \{1, \ldots, m\} : \pi^{Z_j}(a) \in A_j\}$ is an attractor in $S(f)$. Moreover, every attractor in $S(f)|_M$ can be represented in this manner as product of attractors in $S(f_{Z_j})$, $j \in \{1, \ldots, m\}$, and component values $M_i$ for $i \notin J(M)$.*

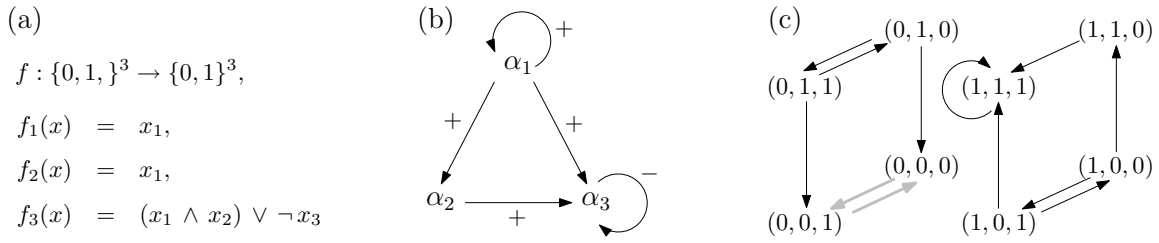



Figure 3: A network with input layer in (a), its global interaction graph in (b), and its state transition graph in (c).

We illustrate the results on our running example from Fig. 1. The graph $G^\theta(M)$ for $M = ([1,2], 1, [0,1])$ has two components, $Z_1$ consisting of a positive loop on $\alpha_1$, and $Z_2$ being a negative loop on $\alpha_3$, as can be seen in Fig. 2 (b). We define $f^{Z_1} : [1,2] \to [1,2]$ as in Def. 4.4 with $\pi^{Z_1}((M_1', M_2', M_3')) = M_1'$ for all $M' \subseteq M$ and $\rho^{Z_1}(z) = (z, 1, [0,1])$ for all $z \in [1,2]$. This reduces the function $f_1$ given in Fig. 1 (a) to $f^{Z_1}(z) = z$. The state transition graph $S(f^{Z_1})$ is given in Fig. 2 (c) and consists of two steady states. Analogously, we obtain $S(f^{Z_2})$, which comprises a single attractor of cardinality 2 as shown in Fig. 2 (d). Applying Def. 4.5 we obtain the graph $S^M$ which is shown in Fig. 2 (e). Comparison with the state transition graph $S(f)$ given in Fig. 1 (c) illustrates Theorem 4.6 and its corollary.

## 5 Networks with input layer

In this section we introduce a class of networks, for which we can easily find a set of symbolic steady states such that all attractors of the original system can be constructed from the attractors of the network modules derived from the symbolic steady states. We extend results obtained in [17].

**Definition 5.1.** *We call $f$ a* network with input layer*, if there exists $i \in \{1, \ldots, n\}$ such that $f_i = \mathrm{id}_{X_i}$. A vertex satisfying this condition is called* input vertex*.*

For every input vertex $\alpha_i$, we have no incoming edges except a positive loop in the global interaction graph. However, this structural criterion is not sufficient for identifying an input vertex. Networks with input layer are well-suited for modeling, for example, signal transduction networks, where receptors may be modeled as input vertices.

In the following we assume that $f$ is a network with input layer. Without loss of generality we assume that $\alpha_1, \ldots, \alpha_k$, $k \in \{1, \ldots, n\}$, are the input vertices of $f$.

We can immediately note one important property of $f$. The frozen core of a symbolic state $M' \in X^\square$ with $M_i' \in X_i$ for $i \in \{1, \ldots, k\}$ and $M_i' = X_i$ for $i > k$ is given by $(\{1, \ldots, k\}, M')$, since $M'$ is its own extended orbit. In other words, we can easily derive a (symbolic) steady state from each combination of values for the input vertices using Theorem 3.3. The resulting set of symbolic and possibly regular steady states is sufficient to determine all attractors of $f$.

**Theorem 5.2.** *Let $A$ be an attractor of $f$. Then there exist input values $x_i \in X_i$, $i \in \{1, \ldots, k\}$ such that either $A = M$ is a regular steady state, or we can construct $A$ from $M$ as shown in*

*Cor 4.7, where $M$ is the fixed point derived from $(\{1,\dots,k\}, M')$, $M' \in X^{\square}$ with $M'_i = x_i$ for $i \in \{1,\dots,k\}$ and $M'_i = X_i$ for $i > k$.*

*Proof.* Since $\alpha_1 \dots \alpha_k$ are input vertices, we have $a_i = a'_i$ for all $i \in \{1,\dots,k\}$, and we set $M'_i = a_i$ for the input vertices. Next, we show that for every state $x \in X$ with $x_i = M'_i$ for all $i \in \{1,\dots,k\}$, there exists a trajectory leading to $M$.

Let $(M^l)_{l\in\mathbb{N}_0}$ with $M^0 := \widetilde{M'}$ and $M^l := F(M^{l-1})$ for all $l \in \mathbb{N}$ be the sequence converging to $M$ introduced in Theorem 3.3. Recall that $M^l \subseteq M^{l-1}$ for all $l \in \mathbb{N}$. Let $x^0 \in M' \setminus M$. Then there exists $l \in \mathbb{N}_0$ such that $x^0 \in M^l \setminus M^{l+1}$. Then, according to the definition of $F$ and since $(M^l)_{l\in\mathbb{N}_0}$ is decreasing with respect to the subset relation, either all successors of $x^0$ in $S(f)$ are in $M^l \setminus M^{l+1}$ or there exists a successor in $M^{l+1}$. In the latter case, we label that successor $x^1$. Otherwise, we repeat the procedure for all successor of $x^0$, check again and if necessary repeat again. Since all images of all states in $M^l$ lie in $M^{l+1}$ and since the state space is finite, we eventually find a state $x^1 \in M^{l+1}$ such that there exists a path from $x^0$ to $x^1$ with all states of the path except $x^1$ lying in $M^l \setminus M^{l+1}$. Since $(M^l)_{l\in\mathbb{N}_0}$ is converging to $M$ we can thus construct a path from each state in $M'$ to $M$.

It follows that there is no trap set, and thus no attractor, in $M' \setminus M$. If $M$ is a regular steady state, then $M = A$ is the only attractor in $M'$. Otherwise $A$ is a composition of attractors of network modules derived from $M$ and the frozen components of $M$ as shown in Cor 4.7. □

We illustrate the results on the simple Boolean network given in Fig. 3 which has one input vertex, namely $\alpha_1$. We start calculating the two (symbolic) steady states from the input values $x_1 = 0$ and $x_1 = 1$. In the first case, we get $F((0,[0,1],[0,1])) = (0,0,[0,1])$ and $F((0,0,[0,1])) = (0,0,[0,1])$, i. e., $(0,0,[0,1])$ is a symbolic steady state, and the associated structural module is a negative loop originating in $\alpha_3$. When looking at the corresponding network module, we obtain the function $f^Z : [0,1] \to [0,1]$, $f^Z(z) = \neg\, z$, and the derived attractor for the original network is the set $\{(0,0,0),(0,0,1)\}$. For the input value $x_1 = 1$ we get the sequence $((1,[0,1],[0,1]),(1,1,[0,1]),(1,1,1),(1,1,1),\dots)$, so the procedure renders a regular steady state of the system. As can be seen in Fig. 3, the regular steady state and the attractor derived from $(0,0,[0,1])$ are the only attractors of the system.

## 6 Analyzing Th cell differentiation

T helper cells, short Th cells, are important players in the vertebrate immune system. They can be sub-classified in Th1 and Th2 cells, which are involved in different immune responses. Both originate from a common precursor, promote their own differentiation and inhibit proliferation of each other. In [10] L. Mendoza proposes a model for a control network of Th cell differentiation consisting of 17 components, 13 of which are represented by Boolean variables while the remaining 4 components have three activity levels. The logical rules governing the behavior of the system are given in Table 1 and the global interaction graph can be seen in Fig. 4. Note that the model depicted in Fig. 4 and Table 1 differs slightly from the model introduced in [10], namely we altered the logical functions associated with the vertices IFN-$\beta$, IL-12 and Il-18. In Mendoza's model, the three components are modeled with constant functions with value 0, representing the wild type in some sense (see [10]). The constant values are changed when considering specific artificial environmental conditions. We model these vertices as input vertices and consider for the wild type the situation where all input values are zero. Clearly,

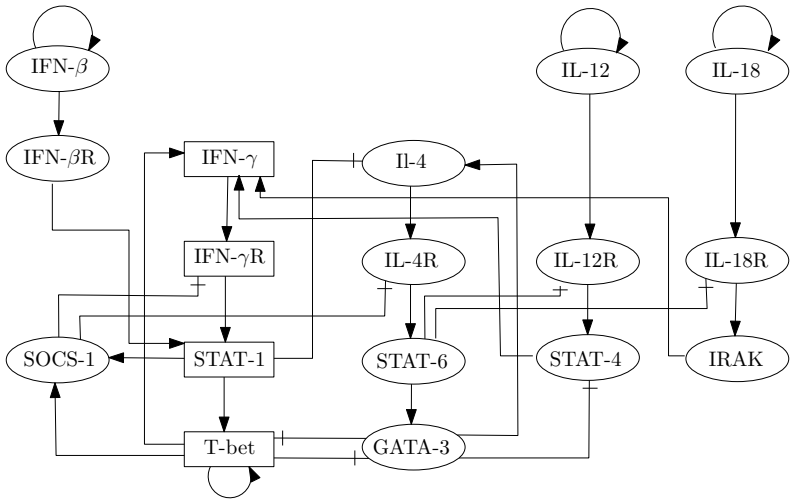



Figure 4: Global interaction graph of the Th cell differentiation network introduced in [10]. Arrows represent activation, crossed lines represent inhibition.

the attractors of both models coincide. Modeling IFN-$\beta$, IL-12 and IL-18 as input vertices also makes sense from a biological point of view since all three vertices represent substances not reproduced by Th cells. If we want to mimic experimental conditions where cells are cultured in media saturated with one or more of these substances, we can easily do so by focussing on the part of state space where one or more of the input vertices have value one. Note that we have no further input vertices in our model.

For the wild type, i. e. the situation where all input values are set to zero, Mendoza identifies four attractors all of which are fixed points of the function $f$ given in Table 1. Each one has a clear biological interpretation [10]. We now want to apply our analysis technique using symbolic steady states to the wild type.

We fix the values of the input vertices to zero and as a first step determine the corresponding symbolic steady state, that is, the symbolic steady state derived from the frozen component set ($\{1,2,3\}$, $x_1 = x_2 = x_3 = 0$). Iterating the state

$$M^0 := (0,0,0,[0,1],[0,2],[0,1],[0,2],[0,1],[0,1],[0,1],[0,1],[0,2],[0,1],[0,1],[0,1],[0,2],[0,1])$$

we get

$$\begin{aligned} M^1 :=&\ f(M^0) = \ (0,0,0,0,[0,2],[0,1],[0,2],[0,1],0,0,[0,1],[0,2],[0,1],[0,1],[0,1],[0,2],[0,1]),\\ M^2 :=&\ f(M^1) = \ (0,0,0,0,[0,2],[0,1],[0,2],[0,1],0,0,[0,1],[0,2],[0,1],0,0,[0,2],[0,1]),\\ &f(M^2) = \ M^2. \end{aligned}$$

We obtain a symbolic steady state with 8 regular components, and no further constraints on the remaining components. The local interaction graph $G^\theta(M)$ is shown in Fig. 5 (a). Analysis of the corresponding subnetwork renders four fixed points, namely

$$(0,0,0,0,0,0,0,0,0)\,,\ (1,0,1,0,1,1,0,1,0)\,,\ (2,0,1,0,1,1,0,2,0)\,,\ (0,1,0,1,0,0,1,0,1)$$

| | | |
|---|---|---|
| IFN-$\beta$ | $X_1 = \{0,1\}$ | $f_1(x) = x_1$ |
| IL-12 | $X_2 = \{0,1\}$ | $f_2(x) = x_2$ |
| IL-18 | $X_3 = \{0,1\}$ | $f_3(x) = x_3$ |
| IFN-$\beta$R | $X_4 = \{0,1\}$ | $f_4(x) = x_1$ |
| IFN-$\gamma$ | $X_5 = \{0,1,2\}$ | $f_5(x) = 1$ if $(x_{16} = 1 \wedge \neg(x_{14} = 1 \wedge x_{15} = 1)) \vee (x_{14} = 1 \wedge x_{15} = x_{16} = 0)$, $f_5(x) = 2$ if $x_{16} = 2 \vee (x_{14} = 1 \wedge x_{15} = 1)$, and $f_5(x) = 0$ otherwise |
| IL-4R | $X_6 = \{0,1\}$ | $f_6(x) = 1$ if $x_{12} = 0 \wedge x_{17} = 1$, and $f_6(x) = 0$ otherwise |
| IFN-$\gamma$R | $X_7 = \{0,1,2\}$ | $f_7(x) = 1$ if $x_5 = 1 \vee (x_5 = 2 \wedge x_{11} = 1)$, $f_7(x) = 2$ if $x_5 = 2 \wedge x_{11} = 0$, and $f_7(x) = 0$ otherwise |
| IL-4R | $X_8 = \{0,1\}$ | $f_8(x) = x_6 \wedge \neg x_{11}$ |
| IL-12R | $X_9 = \{0,1\}$ | $f_9(x) = x_2 \wedge \neg x_{13}$ |
| IL-18R | $X_{10} = \{0,1\}$ | $f_{10}(x) = x_3 \wedge \neg x_{13}$ |
| SOCS-1 | $X_{11} = \{0,1\}$ | $f_{11}(x) = 1$ if $x_{12} \geq 1 \vee x_{16} \geq 1$, and $f_{11}(x) = 0$ otherwise |
| STAT-1 | $X_{12} = \{0,1,2\}$ | $f_{12}(x) = 1$ if $(x_4 = 1 \wedge x_7 = 0) \vee x_7 = 1$, $f_{12}(x) = 2$ if $x_7 = 2$, and $f_{12}(x) = 0$ otherwise |
| STAT-6 | $X_{13} = \{0,1\}$ | $f_{13}(x) = x_8$ |
| STAT-4 | $X_{14} = \{0,1\}$ | $f_4(x) = x_9 \wedge \neg x_{17}$ |
| IRAK | $X_{15} = \{0,1\}$ | $f_{15}(x) = x_{10}$ |
| T-bet | $X_{16} = \{0,1,2\}$ | $f_{16}(x) = 1$ if $(x_{17} = 0 \wedge ((x_{12} = 1 \wedge x_{16} \leq 1) \vee (x_{12} \leq 1 \wedge x_{16} = 1))) \vee (x_{17} = 1 \wedge x_{16} = 1 \wedge x_{12} = 1)$, $f_{16} = 2$ if $(x_{17} = 0 \wedge (x_{12} = 2 \vee x_{16} = 2)) \vee (x_{17} = 1 \wedge x_{12} = 1 \wedge x_{16} = 2)$, $f_{16}(x) = 0$ otherwise |
| GATA-3 | $X_{17} = \{0,1\}$ | $f_{17}(x) = 1$ if $x_{13} = 1 \wedge x_{16} = 0$, and $f_{17}(x) = 0$ otherwise |

Table 1: Coordinate functions and ranges for the components of the Th cell network.

in the subspace $X_5 \times X_6 \times X_7 \times X_8 \times X_{11} \times X_{12} \times X_{13} \times X_{16} \times X_{17}$. The steady states of the original network derived from these fixed points match the four steady states found in [10]. The state space of the original model consists of 663552 states. Fixing the input values still leaves us with 82944 states to consider. The state space of the structural module associated with the symbolic steady state $M^2$ contains only 2592 states.

Not all of the combinations of input values render a significant simplification of the network analysis. In the worst case, for example if we choose input values $x_1 = 0$, $x_2 = x_3 = 1$, we can only derive the value for $x_4$ but no further constraints on structure and behavior of the system. This in itself is of course an interesting observation from a biological point of view, since in that case we can deduce that cross-regulation plays an important role at an early stage of signal transduction.

On the other hand, some combinations of input values lead to very small network modules. Let us as a last example consider the input values $x = 1$, $x_2 = x_3 = 0$, representing an overabundance of IFN-$\beta$.

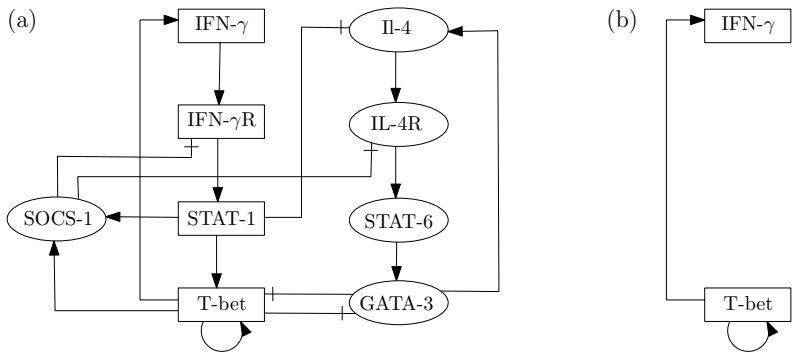



Figure 5: Subnetworks of the Th cell differentiation network associated with the symbolic fixed points derived from the input values $x_1 = x_2 = x_3 = 0$ in (a) and $x_1 = 1$, $x_2 = x_3 = 0$ in (b).

Starting with

$$M^0 := (1, 0, 0, [0,1], [0,2], [0,1], [0,2], [0,1], [0,1], [0,1], [0,1], [0,2], [0,1], [0,1], [0,1], [0,2], [0,1])$$

we get

$$\begin{aligned}
M^1 &:= f(M^0) = (1, 0, 0, 1, [0,2], [0,1], [0,2], [0,1], 0, 0, [0,1], [0,2], [0,1], [0,1], [0,1], [0,2], [0,1]),\\
M^2 &:= f(M^1) = (1, 0, 0, 1, [0,2], [0,1], [0,2], [0,1], 0, 0, [0,1], [1,2], [0,1], 0, 0, [0,2], [0,1]),\\
M^3 &:= f(M^2) = (1, 0, 0, 1, [0,2], 0, [0,2], [0,1], 0, 0, 1, [1,2], [0,1], 0, 0, [0,2], [0,1]),\\
M^4 &:= f(M^3) = (1, 0, 0, 1, [0,2], 0, [0,1], 0, 0, 0, 1, [1,2], [0,1], 0, 0, [0,2], [0,1]),\\
M^5 &:= f(M^4) = (1, 0, 0, 1, [0,2], 0, [0,1], 0, 0, 0, 1, 1, 0, 0, 0, [0,2], [0,1]),\\
M^6 &:= f(M^5) = (1, 0, 0, 1, [0,2], 0, [0,1], 0, 0, 0, 1, 1, 0, 0, 0, [0,2], 0),\\
M^7 &:= f(M^6) = (1, 0, 0, 1, [0,2], 0, [0,1], 0, 0, 0, 1, 1, 0, 0, 0, [1,2], 0),\\
M^8 &:= f(M^7) = (1, 0, 0, 1, [1,2], 0, [0,1], 0, 0, 0, 1, 1, 0, 0, 0, [1,2], 0),\\
M^9 &:= f(M^8) = (1, 0, 0, 1, [1,2], 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, [1,2], 0),\ f(M^9) = M^9.
\end{aligned}$$

We obtain a two-component module consisting of a positive loop on T-bet and an activating edge from T-bet to IFN-$\gamma$ which can be seen in Fig. 5 (b). Both components of the module originally have three activity levels, but are both constrained to levels 1 and 2 in the module dynamics. Thus, we only have to analyze a state space of cardinality four instead of a state space consisting of 82944 states. The module has two steady states, namely $(x_5, x_{16}) = (1, 1)$ and $(x_5, x_{16}) = (2, 2)$, which translate to two steady states in the original network. Again, this is in agreement with the results in [10] (supplementary material).

Application of our analysis method to this model thus offers two advantages. First, the complexity of the analysis of the dynamics is reduced, since we only have to focus on the smaller network modules. Furthermore, identification of the modules themselves is of interest, since they represent the part of the system responsible for the decision of the system's fate, i. e., which attractor is reached. An interesting next step would then be to check whether the mathematically derived network modules coincide with subsystems of known biological importance.

# 7 Conclusion

In this paper we introduced a method to analyze discrete regulatory networks utilizing suitable network modules. We used the notion of symbolic steady state, which allows us to determine such network modules using coarse dynamical and subsequently structural modules derived from the symbolic steady state. Lastly, we can associate network modules with the structural modules exploiting the properties of the frozen core of the symbolic steady state. We then can construct the dynamics of the original network, and in particular its attractors, in a subset of state space explicitly from the state transition graphs of the network modules. This paper not only gives a rigorous definition of different aspects of modularity but notably extends results in [17, 16]. In particular, the detailed analysis of the Th cell network becomes possible because of the refined notion of symbolic steady state.

A variety of aspects provide possibilities for fruitful future work. Firstly, we want to focus on further options for easily computing symbolic steady states. Here, we introduced a method suited for networks with input-layer, but the resulting symbolic steady states might not be minimal with respect to the subset relation on symbolic steady states, and thus the resulting network modules can possibly be further refined. We anticipate results when we focus on certain classes of functions $f$ describing networks, in particular (nested) canalyzing functions [8, 7]. A different direction of interest is to use the network modules not only for obtaining the system's dynamics but also for a refined stability analysis. Perturbations resulting in changes in the dynamics of $f$ might not be noticeable in every network module (see [6]). Such considerations are of similar interest when using the synchronous update strategy, which is also often utilized in discrete modeling, so a translation of our results to synchronous update networks seems worthwhile. Furthermore, we need to compare our results to other well-established modularization techniques that aim at reducing the analysis complexity such as the modular response analysis of biochemical networks [9, 2]. Although the underlying modeling approaches are very different, some ideas may be transferable or render complementing results. Lastly, we plan to apply our methods to further biological examples. Here, a comparison of the network modules and their associated structural and dynamical modules with subsystems of known biological importance could lead to interesting insights.

# References

[1] U. Alon. Network motifs: theory and experimental approaches. *Nat. Rev. Genet.*, 8:450–461, 2007.

[2] F. J. Bruggeman, H. V. Westerhoff, J. B. Hoek, and B. N. Kholodenko. Modular response analysis of cellular regulatory networks. *J. Theoret. Biol.*, 218(4):507 – 520, 2002.

[3] C. Christensen and R. Albert. Using graph concepts to understand the organization of complex systems. *IJBC*, 17(7):2201–2214, 2007.

[4] M. Ederer, T. Sauter, E. Bullinger, E. Gilles, and F. Allgwer. An approach for dividing models of biological reaction networks into functional units. *Simulation*, 79(12):703 – 716, 2003.

[5] L. Hartwell, J. Hopfield, S. Leibler, and A. Murray. From molecular to modular cell biology. *Nature*, 402:C47 – C52, 1999.

[6] D. Irons and N. Monk. Identifying dynamical modules from genetic regulatory systems: applications to the segment polarity network. *BMC Bioinformatics*, 8(1):413, 2007.

[7] A. S. Jarrah and R. Laubenbacher. Discrete models of biochemical networks: the toric variety of nested canalyzing functions. In *Algebraic Biology, AB 2007, Castle of Hagenberg, Austria*, volume 4545 of *LNCS*, pages 15–22. Springer, 2007.

[8] S. Kauffman. *The Origins of Order*. Oxford University Press, 1993.

[9] B. N. Kholodenko, S. Schuster, J. M. Rohwer, M. Cascante, and H. V. Westerhoff. Composite control of cell function: metabolic pathways behaving as single control units. *FEBS Letters*, 368(1):1 – 4, 1995.

[10] L. Mendoza. A network model for the control of the differentiation process in th cells. *Biosystems*, 84(2):101 – 114, 2006.

[11] M. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci.*, 103(23):8577 – 8582, 2006.

[12] J. A. Papin, J. L. Reed, and B. O. Palsson. Hierarchical thinking in network biology: the unbiased modularization of biochemical networks. *Trends in Biochemical Sciences*, 29(12):641 – 647, 2004.

[13] E. Remy, P. Ruet, and D. Thieffry. Graphic requirements for multistability and attractive cycles in a boolean dynamical framework. *Adv. Appl. Math.*, 41(3):335–350, 2008.

[14] A. Richard. Positive circuits and maximal number of fixed points in discrete dynamical systems. *Discrete Appl. Math.*, 157:3281 – 3288, 2009.

[15] F. Robert. *Discrete Iterations: A Metric Study*, volume 6 of *Springer Series in Computational Mathematics*. Springer, 1986.

[16] H. Siebert. Analysis of discrete bioregulatory networks using symbolic steady states. Preprint 555, DFG Research Center MATHEON, 2009. `http://www.matheon.de/download.asp?File=5576_Siebert_1_09.pdf`.

[17] H. Siebert. Deriving behavior of boolean bioregulatory networks from subnetwork dynamics. *Math. Comput. Sci.*, 2(3):421–442, 2009.

[18] R. Thomas and R. d'Ari. *Biological Feedback*. CRC Press, 1990.

[19] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. II. Logical analysis of regulatory networks in terms of feedback circuits. *Chaos*, 11:180–195, 2001.

# Part III

# Refinement

# Chapter 6

# Comparing Discrete and Piecewise Affine Differential Equation Models of Gene Regulatory Networks

Shahrad Jamshidi and Heike Siebert.

**Remarks.** Shahrad Jamshidi is a doctoral student supervised by Alexander Bockmayr and myself, who works on the relations between discrete and hybrid discrete/continuous modeling formalisms. This chapter focusses on comparing the directed graphs representing the dynamics of a Thomas model and the qualitative dynamics of a piecewise affine differential equation model. Shahrad Jamshidi did most of the preliminary work for this paper, conjecturing and partially proving some correspondences between the respective transition graphs. He also constructed the counter examples presented in this chapter. I developed and proved the theorem clarifying the local relation between the two graphs, focussing on a given vertex, the adjacent vertices, and the edges between them. Shahrad Jamshidi and I then used the theorem to prove the correspondence of some global dynamical properties. This chapter consists of a first draft of our findings, the results have not been published yet.

# Comparing Discrete and Piecewise Affine Differential Equation Models of Gene Regulatory Networks

Shahrad Jamshidi and Heike Siebert

*Abstract.* Mathematical modelling often helps to provide a systems perspective on gene regulatory networks. In particular, qualitative methods are useful when kinetic information is lacking. Multiple methods have been developed that implement qualitative information in different ways, e.g., in purely discrete or hybrid discrete/continuous models. In this paper, we compare the discrete formalism of R. Thomas with a modeling formalism based on piecewise affine differential equations. Respective models in the two formalisms carry essentially the same information, however, the latter formalism explicitly includes threshold values for the network interactions. We provide a local characterization of the qualitative dynamics of a piecewise affine differential equation model using the discrete dynamics of a corresponding Thomas model. Based on this result, we investigate the consistency of higher-level dynamical properties such as attractor characteristics and reachability properties.

## 1 Introduction

Gene regulation is the result of the complicated interplay of many molecular components forming large and complex interaction networks. Mathematical modelling of such gene regulatory networks often allows insights into the underlying structure and dynamics of various biological systems. If information on kinetic parameters is lacking, qualitative formalisms offer a well-established alternative to the more traditionally used differential equation models. They incorporate the information inherent in the network structure and coarse information pertaining the interactions between components, resulting in a rather abstract description of the system's dynamics.

The discrete formalism introduced by R. Thomas is a qualitative method describing a gene regulatory network by a discrete function. Each network component is represented by a variable that takes integer values representing the different levels of gene activity. The information on how the behavior of the component is governed by the values of the other network components is captured in a discrete function. The component functions then constitute the coordinate functions of the update function of the network. To derive the dynamics of the system, Thomas introduced the asynchronous update method where only one variable changes per discrete time step, and only by a unit value. Since the state space is finite, the dynamics can be represented by a directed graph, the so-called asynchronous state transition graph [12].

The particularities of the asynchronous update method result in a close correspondence of the discrete model to certain differential equation systems [10]. Other qualitative modeling

approaches are even more obviously related to continuous models. Differential equation models consisting of step functions retain a continuous time evolution, yet can be seen as qualitative due to the close relation of step and discrete functions. Such piecewise affine differential equation (PADE) models approximate certain ordinary differential equation models [6, 7]. H. de Jong and colleagues have shown that they can essentially be captured by a discrete representation which abstracts the continuous solution trajectories of the differential equations into transitions between different regions of the phase space [3]. Again, the resulting dynamics can be represented by a directed graph, the so-called qualitative transition graph.

In this paper, we aim at clarifying the relation between Thomas and PADE models by comparing the respective graphs capturing the dynamical behavior. Several results in this direction already exist. In particular, attractors, including steady states and certain limit cycles, are related [6, 10, 11, 4, 2, 14]. However, in all cases the focus is on some specific dynamical property. Here, we present a comprehensive comparison between the state transition and the qualitative transition graph. Initially, we only aim at an understanding on the local level, i.e., focussing on the outgoing edges of a given vertex. The results are then applied to more complex structures of the graphs and global characteristics, in agreement with the previous work cited above.

The paper is organized as follows. Section 2 introduces a well-established discrete modeling approach based on the Thomas formalism. PADE systems and the qualitative analysis developed by de Jong and colleagues are introduced in Sect. 3. In the following section, we clarify the relation between the two formalisms. Section 5 contains our main result characterizing transitions in the qualitative transitions graphs using edges originating in corresponding vertices in the state transition graph. We illustrate the application of this result by establishing relations between paths and attractors in the two graphs. A short section including final remarks and some perspectives for future work concludes the paper.

## 2 Discrete formalism

Consider a gene network with $n$ regulatory components. In the discrete modeling approach, the activity level of component $i \in \{1, \ldots, n\}$ is modeled by a discrete variable $q_i$, which takes its values in a finite set of natural numbers $Q_i = \{0, \ldots, p_i\}$. The *state space* of the discrete model is $Q = Q_1 \times \cdots \times Q_n$ and the regulatory interactions are captured by a discrete *update function* $f = (f_1, \ldots, f_n) : Q \to Q$.

Often in discrete modeling, an *interaction graph* with vertex set $\{1, \ldots, n\}$ and edges representing interactions between components is used to visualize the network structure. This paper, however, is focussed solely on the dynamics of a network, so that we refrain from giving a formal definition.

From $f$ we obtain the *state transition graph* $STG(f) = (Q, E)$, which is a directed graph with node set $Q$ and edge set $E \subset Q \times Q$. For any $j \in \{1, \ldots, n\}$ with $f_j(q) \neq q_j$ there is an edge $(q, q') \in E$, where

$$q'_j = q_j + sgn(f_j(q) - q_j) \text{ and } q'_i = q_i, \text{ for all } i \in \{1, \ldots, n\} \setminus \{j\}.$$

Here, $sgn : \mathbb{R} \to \{-1, 0, 1\}$ denotes the sign function. If $f(q) = q$, then $(q, q) \in E$ and $q$ is called a *fixed point*.

| $q$ | $f(q)$ |
|---|---|
| 00 | 12 |
| 01 | 12 |
| 02 | 11 |
| 10 | 00 |
| 11 | 10 |
| 12 | 11 |

| $q$ | $f^G(q)$ |
|---|---|
| 00 | 11 |
| 01 | 12 |
| 02 | 11 |
| 10 | 00 |
| 11 | 10 |
| 12 | 11 |

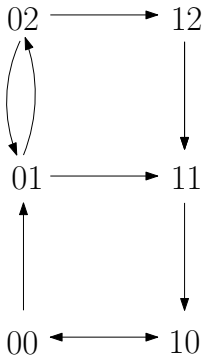



Figure 1: Example network consisting of two components. On the left are given the function values of the update function as well as of the unitary update function, on the right the corresponding state transition graph $G$.

The update function uniquely determines the state transition graph. Unless $f$ is Boolean, it is not possible to recover $f$ from $G = STG(\mathrm{f})$. However, we may obtain from $G$ a *unitary update function* $f^G : Q \to Q$ by setting

$$f_j^G(q) = q_j + \sum_{q' \in AS(q)} (q'_j - q_j), \text{ for } j \in \{1, \ldots, n\}.$$

Here $AS(q) := \{q' \in Q \mid (q, q') \in E\}$ denotes the set of *asynchronous successors* of $q$ in $G$.

**Lemma 2.1.** *Let $f : Q \to Q$ be an update function and $G = STG(f)$. Then*

$$STG(f) = STG(f^G).$$

*Proof.* Let $j \in \{1, \ldots, n\}$ and $q \in Q$. By definition of $AS(q)$, there exists at most one $q' \in AS(q)$ such that $|q'_j - q_j| = 1$. This implies $sgn(f_j(q) - q_j) = q'_j - q_j$. Therefore, $sgn(f_j(q) - q_j) = \sum_{q' \in AS(q)} (q'_j - q_j)$, and the result follows. □

The unitary update function $f^G$ captures the information from the original update function $f$ contained in $G = STG(f)$. If $f$ is Boolean, $f$ and $f^G$ are the same.

The following example will be used throughout the paper.

**Example 2.2.** *For $Q = \{0, 1\} \times \{0, 1, 2\}$, we give an update function $f : Q \to Q$ with state transition graph $G = STG(f)$ and corresponding unitary update function $f^G$ as specified in Fig. 1.*

## 3 Piecewise affine differential equations

In the following, we describe a formalism using piecewise affine differential equations (PADEs) to represent a regulatory network together with a qualitative analysis approach for such systems. The method was introduced by de Jong and colleagues, details can be found in [3]. There have

been a number of refinements proposed for the approach, however, we chose to use the original approach for the comparison with the purely discrete formalism. The results presented here may then be expanded for the more refined PADE formalism.

The focus of this section is the *qualitative dynamics* associated with a system of PADEs, i.e., the abstraction of all possible solution trajectories of the PADEs satisfying certain parameter constraints.

Consider an $n$-dimensional phase space $\Omega = \Omega_1 \times \cdots \times \Omega_n \subset \mathbb{R}^n_{\geq 0}$, where

$$\Omega_i = \{x_i \in \mathbb{R} \mid 0 \leq x_i \leq max_i\},$$

and $max_i \in \mathbb{R}_{>0}$. For every continuous variable $x_i \in \Omega_i$ we assume there are $p_i \in \mathbb{N}$ thresholds $\theta_i^1, \cdots, \theta_i^{p_i}$, which satisfy the threshold order

$$0 < \theta_i^1 < \cdots < \theta_i^{p_i} < max_i, \quad \text{for all } i \in \{1, \ldots, n\}. \tag{1}$$

Since we aim at a comparison between a system described in this formalism with a discrete system as introduced in the previous section, the value $p_i$ chosen here corresponds to the maximal value $p_i$ of the component range $Q_i$ of a discrete model.

We consider a set of PADEs in $\Omega$ of the form

$$\dot{x}_i = F_i(x) - G_i(x)x_i, \quad i \in \{1, \ldots, n\}, \tag{2}$$

where the functions $G_i : \Omega \to \mathbb{R}_{>0}$ and $F_i : \Omega \to \mathbb{R}_{\geq 0}$ are linear combinations of products of step functions

$$S^+(x_l, \theta_l^k) = \begin{cases} 0 & \text{if } x_l < \theta_l^k, \\ 1 & \text{if } x_l > \theta_l^k, \end{cases}$$

and $S^-(x_l, \theta_l^k) = 1 - S^+(x_l, \theta_l^k)$ for $l \in \{1, \ldots, n\}$. Note that the differential equations remain undefined for threshold values.

To obtain a qualitative representation of the PADE system, the state space is partitioned into a set of domains.

**Definition 3.1.** *Consider a set of PADEs of the form (2) with phase space $\Omega$ and thresholds $\theta_i^j$. The $(n-1)$-dimensional hyperplanes corresponding to the equations $x_i = \theta_i^j$, $j \in \{1, \ldots, p_i\}$, divide $\Omega$ into hyperrectangular regions called* domains. *A domain $D \subset \Omega$ is defined by $D = D_1 \times \cdots \times D_n$ where every $D_i$ is given by one of the following equations*

$$\begin{aligned} D_i &= \{x_i \,|\, 0 \leq x_i < \theta_i^1\}, \\ D_i &= \{x_i \,|\, \theta_i^k < x_i < \theta_i^{k+1}\} \text{ for } k \in \{1, \ldots, p_i - 1\}, \\ D_i &= \{x_i \,|\, \theta_i^{p_i} < x_i \leq max_i\}, \\ D_i &= \{x_i \,|\, x_i = \theta_i^k\} \text{ for } k \in \{1, \ldots, p_i\}. \end{aligned}$$

*By $\mathcal{D}$ we denote the set of all domains in $\Omega$.*

*A domain $D \in \mathcal{D}$ is called a* switching domain, *if there exists $i \in \{1, \ldots, n\}$ such that $D_i = \{x_i \,|\, x_i = \theta_i^k\}$ for some $k \in \{1, \ldots, p_i\}$. The variable $x_i$ is then called* switching variable. *The* order *of a switching domain is the number of its switching variables. A domain $D \in \mathcal{D}$ is called a* regulatory domain, *if it is not a switching domain.*

*The set of regulatory and switching domains are denoted by $\mathcal{D}_r$ and $\mathcal{D}_s$, respectively.*

It follows immediately that for any regulatory domain $D \in \mathcal{D}_r$, the functions $F_i(x)$ and $G_i(x)$ are constant for all $x \in D$. Thus (2) can be written as a linear vector field

$$\dot{x} = F^D - G^D x, \quad x \in D,$$

where $G^D = diag(G_1^D, \ldots, G_n^D)$ is a diagonal matrix of strictly positive numbers and $F^D = (F_1^D, \ldots, F_n^D)$ is a vector of positive numbers.

The PADEs are not defined on the threshold hyperplanes, and it is difficult to define suitable dynamics of (2) on switching domains. To deal with this problem the differential equations are extended to differential inclusions, and methods presented in [5], [8] and [3] are applied to obtain so-called Fillipov solutions of the PADE. However, our focus is on the qualitative dynamics, the definition of which does not depend on the particularities of the Fillipov extension. For more details, we refer the reader to the cited literature and references therein.

The qualitative analysis on the switching domains requires the following notions.

**Definition 3.2.** *Consider a set of PADEs of the form (2) with domain set $\mathcal{D}$. For every $D \in \mathcal{D}_s$ of order $k$, let $supp(D)$ be the $(n-k)$-dimensional hyperplane in $\Omega$ containing $D$. If $D \in \mathcal{D}_r$, then $supp(D) = \Omega$. The* boundary *of $D$ in $supp(D)$ is the set $\partial D$ of all points $x \in supp(D)$, such that each ball $B_D(x, \epsilon)$ in $supp(D)$ of center $x$ and radius $\epsilon > 0$ intersects both $D$ and $supp(D) \backslash D$. For all $D \in \mathcal{D}_s$, we define the set*

$$R(D) = \{D' \in \mathcal{D}_r | D \subset \partial D'\}.$$

So, $R(D)$ is the set of all regulatory domains that have $D$ in their boundary. With this relation of switching domains to multiple regulatory domains we can extend the dynamics of the regulatory domains to the switching domains.

**Definition 3.3.** *Consider a set of PADEs with domain set $\mathcal{D}$. We define the* focal set $\Psi(D)$ *for every domain $D$ as follows:*

$$\Psi(D) := \begin{cases} \{\phi(D)\} & \text{if } D \in \mathcal{D}_r, \\ supp(D) \cap \overline{rect}(\{\phi(D') | D' \in R(D)\}) & \text{if } D \in \mathcal{D}_s, \end{cases}$$

*where $\phi(D) := (G^D)^{-1} F^D$ is called the* focal point of $D$ *for $D \in \mathcal{D}_r$ and $\overline{rect}(P)$ is the smallest closed hyperrectangular set that contains $P \subset \Omega$.*

In the following, we always assume certain parameter constraints in agreement with the approach in [3], namely, we assume that all focal points lie in a regulatory domain. Then, by definition of the regulatory domains, we can encode the position of each focal point by strict inequalities using the threshold values, and thus obtain a parameter constraint of the form of (1) consisting of threshold values and components of the focal point vector. We call these constraints *ordering constraints*.

It is easy to see, that solutions of (2) starting in a regulatory domain $D$ converge monotonically towards $\phi(D)$. Using again the Fillipov extensions, the behavior of the system in switching domains can be described in relation to the focal points of adjacent regulatory domains. Using a hyperrectangular set in defining the $\Psi(D)$ for switching domains results in an overapproximation of the PADE behavior by the qualitative behavior we introduce below. The consequences of this overapproximation are discussed in detail in [3].

From [7], e.g., we know that solution trajectories of (2) starting in a domain $D$ converge monotonically to the (non-empty) focal set $\Psi(D)$. Because we focus on the qualitative dynamics, the possible behaviors in a domain $D$ can therefore be determined by the relative position of the focal set $\Psi(D)$ with respect to the domain $D$.

**Definition 3.4.** *Consider a set of PADEs of the form (2). Let $D \in \mathcal{D}$ and $e \in \Omega$. We call the mapping $v : \mathcal{D} \times \Omega \to \{-1, 0, 1\}^n$ the* relative position vector *and define it as follows*

$$v_i(D, e) = \begin{cases} -1 & \text{if } e_i < x_i, \text{ for all } x \in D, \\ 0 & \text{if } e_i = x_i, \text{ for some } x \in D, \\ +1 & \text{if } e_i > x_i, \text{ for all } x \in D. \end{cases}$$

*Let $E \subset \Omega$ be a non-empty set of points. The set $V(D, E)$ is defined as*

$$V(D, E) := \{v(D, e) | e \in E\}.$$

Taking into account the above considerations about the behavior of the system with respect to the focal points, we can interpret the $i$-th component of $\nu \in V(D, \Psi(D))$ as an order for the variable $x_i$ to increase ($\nu_i = 1$), to decrease ($\nu_i = -1$) or to remain steady ($\nu_i = 0$), in domain $D$. Note that the definition of the domains in $\mathcal{D}$ ensures that $V(D, D')$ is a singleton for all $D, D' \in \mathcal{D}$.

With this idea in mind, we define the qualitative dynamics via transitions between domains.

**Definition 3.5.** *Consider a set of PADEs of the form (2). Let $D, D' \in \mathcal{D}$ such that either $D' \subset \partial D$ or $D \subset \partial D'$. Let $V(D, D') = \{w\}$. We say that there exists a transition from $D$ to $D'$ if one of the following two properties holds:*

1. *If $D' \subset \partial D$,*
   (a) *$V(D, \Psi(D)) \neq \emptyset$ and*
   (b) *there exists $\nu \in V(D, \Psi(D))$ such that $\nu_i w_i = 1$ for every $x_i, i \in \{1, \ldots, n\}$ that is a switching variable in $D'$ but not in $D$.*
2. *If $D \subset \partial D'$,*
   (a) *$V(D', \Psi(D')) \neq \emptyset$ and*
   (b) *there is some $\nu \in V(D', \Psi(D'))$ such that $\nu_i w_i \neq -1$ for every $x_i, i \in \{1, \ldots, n\}$ that is a switching variable in $D$ but not in $D'$.*

Definition 3.5 is extracted from Prop. 6.4 and 6.5 of [3], which characterizes the purely qualitative conditions using a definition for transitions between adjacent domains based on the existence of suitable solution trajectories of (2).

Following the idea of the discrete state transition graph, we summarize all qualitative dynamics of a PADE system in a directed graph.

**Definition 3.6.** *Let $\mathcal{A}$ be a system of PADEs of the form (2). The* qualitative transition graph, *$QTG(\mathcal{A}) = (\mathcal{D}, \mathcal{T})$, is a directed graph with $\mathcal{D}$ being the set of domains and $\mathcal{T}$ the set of transitions between domains.*

It is shown in [3] that all systems in the class of PADEs satisfying the same ordering constraints have the same qualitative dynamics. Transitions in the qualitative state transition graph (QTG) represent possible trajectories of the PADE system, as mentioned. However, the QTG is an overapproximation of the dynamics of the PADE, as discussed in [3].

By the nature of the transition definition, we can easily see that we do not need the information inherent in the ordering constraints in full strength to determine the outgoing transitions for a given domain $D$. Similar to the sufficiency of the unitary update function for the construction of the state transition graph, we can utilize just the set $V(D, \Psi(D))$ for all $D \in \mathcal{D}$ to determine the QTG.

For a regulatory domain $D$ we have

$$V(D, \Psi(D)) = \{v(D, \phi(D))\},$$

by the focal set definition. For a switching domain $D$, the situation is not as clear-cut. However, Prop. 6.2 and 6.3 in [3] characterize the set $V(D, \Psi(D))$ also for switching domains. The characterization is rooted in the overapproximation of the set of focal points of adjacent regulatory domains by a hyperrectangle. The cited results are not formulated in terms of relative postion vectors, but they can easily be rephrased. Thus, we derive the following proposition.

**Proposition 3.7.** *Consider a set of PADEs of the form (2) and let $D \in \mathcal{D}_s$. We have $V(D, \Psi(D)) \neq \emptyset$ if and only if for all switching variables $x_i$ in $D$ we have*

$$\min_{D' \in R(D)} v_i(D, \phi(D')) = -1 \text{ and } \max_{D' \in R(D)} v_i(D, \phi(D')) = 1. \tag{3}$$

*Let $D \in \mathcal{D}_s$ and $V(D, \Psi(D)) \neq \emptyset$. Define $V_i(D, \Psi(D)) := \{\nu_i | \nu \in V(D, \Psi(D))\}$. Then, for all $i \in \{1, \ldots, n\}$, if $x_i$ is a switching variable, $V_i(D, \Psi(D)) = \{0\}$, and if $x_i$ is a non-switching variable*

$$V_i(D, \Psi(D)) = [\min_{D' \in R(D)} v_i(D', \phi(D')), \max_{D' \in R(D)} v_i(D', \phi(D'))], \tag{4}$$

*where $[a, b] = \{a, a+1, \ldots, b-1, b\}$ is the discrete interval for $a, b \in \mathbb{N}$.*

To determine the transitions of the QTG, we only need to know the set $V(D, \Psi(D))$ for all domains $D$. Obviously, we can derive this set immediately from $v(D, \phi(D))$ for regulatory domains $D$. Easy calculation using Prop. 3.7 and the definition of $\Psi(D')$ for switching domains show that the information inherent in the set $\{v(D, \phi(D)) : D \in \mathcal{D}_r\}$ is sufficient to derive $V(D', \Psi(D'))$ for a switching domain $D'$. We summarize the observations in the following lemma.

**Lemma 3.8.** *Let $\mathcal{A}$ be a set of PADEs. The positions of the focal points in relation to their corresponding regulatory domains, i.e., $\{v(D, \phi(D)) : D \in \mathcal{D}_r\}$, is sufficient to calculate $QTG(\mathcal{A})=(\mathcal{D}, \mathcal{T})$.*

In turn, it is easy to see from the definitions that we can derive the relative position vectors $v(D, \phi(D))$ for regulatory domains from the QTG.

To illustrate the notions introduced in this section, we give a short example.

**Example 3.9.** *Consider the following system of PADEs:*

$$\begin{aligned} \dot{x}_1 &= \alpha_1 + \beta_1 S^+(x_1, \theta_1^1) S^-(x_2, \theta_2^1) - \lambda_1 x_1, \\ \dot{x}_2 &= \alpha_2 + \beta_2 S^+(x_1, \theta_1^1) S^-(x_2, \theta_2^2) + \gamma_2 S^-(x_1, \theta_1^1) S^-(x_2, \theta_2^2) - \lambda_2 x_2. \end{aligned}$$

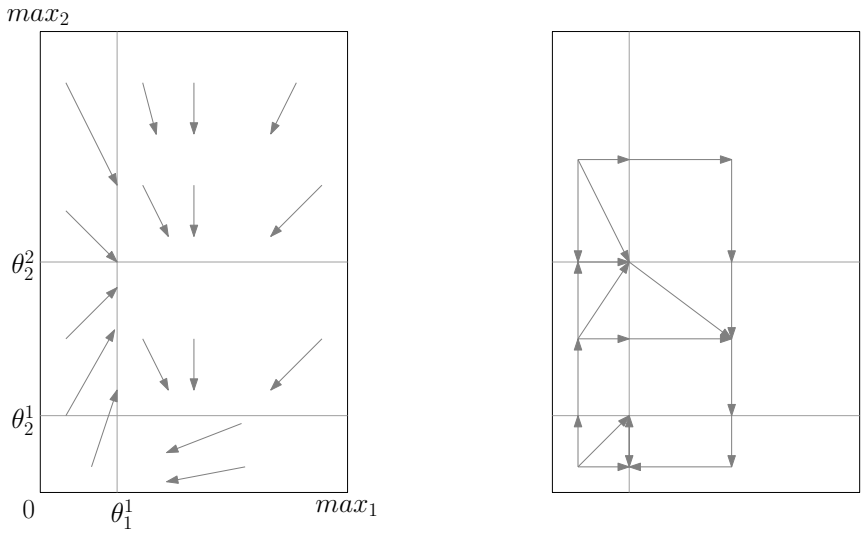



Figure 2: A rough phase portrait of $PADE(f)$ as described in Ex. 3.9 on the left with the corresponding QTG on the right.

*The system has six regulatory domains with corresponding focal points, e.g., the focal point of* $D = [0, \theta_1^1) \times [0, \theta_2^1)$ *being* $(\frac{\alpha_1}{\lambda_1}, \frac{\alpha_2+\gamma_2}{\lambda_2})$. *We impose the ordering constraints*

$$0 < \frac{\alpha_1 + \beta_1}{\lambda_1} < \theta_1^1 < \frac{\alpha_1}{\lambda_1} < max_1 \quad \textit{and} \tag{5}$$

$$0 < \frac{\alpha_2 + \beta_2}{\lambda_2} < \theta_2^1 < \frac{\alpha_2}{\lambda_2} < \theta_2^2 < \frac{\alpha_2 + \gamma_2}{\lambda_2} < max_2. \tag{6}$$

*A phase portrait of the solutions is sketched on the left of Fig. 2. Consider the regulatory domain* $D = [0, \theta_1^1) \times [0, \theta_2^1)$ *and the switching domain of order one* $D' = [\theta_1^1] \times [0, \theta_2^1)$. *We have the corresponding focal sets* $\Psi(D) = \{(\frac{\alpha_1}{\lambda_1}, \frac{\alpha_2+\gamma_2}{\lambda_2})\}$ *and*

$$\Psi(D') = supp(D') \cap \overline{rect}(\{(\frac{\alpha_1}{\lambda_1}, \frac{\alpha_2 + \gamma_2}{\lambda_2}), (\frac{\alpha_1 + \beta_1}{\lambda_1}, \frac{\alpha_2 + \beta_2}{\lambda_2})\})$$

$$= [\theta_1^1] \times [\frac{\alpha_2 + \beta_2}{\lambda_2}, \frac{\alpha_2 + \gamma_2}{\lambda_2}].$$

*The relative positions yield* $V(D, \Psi(D)) = \{(1, 1)\}$ *and* $V(D', \Psi(D')) = \{(0, 1), (0, 0)\}$. *Because* $D' \subset \partial D$ *and* $V(D, D') = \{(1, 0)\}$, *we have* $(D, D') \in \mathcal{T}$. *The complete QTG can be seen on the right of Fig. 2.*

## 4 Relating the discrete and the PADE-based formalisms

In the following we show that the PADE and the discrete formalism are closely related. More specifically, we can obtain a discrete update function from a PADE system via discretization.

In turn, a discrete update function can easily be transformed into a PADE system that shares the qualitative dynamical properties.

A straightforward method to transform a PADE system into a discrete update function was already proposed by Snoussi in [10]. As a first step, the continuous state space of the PADE is discretized according to its threshold values. This leads to the following definition.

**Definition 4.1.** *Let $\mathcal{A}$ be a set of PADEs as in (2), where each variable $x_i$ has $p_i$ ordered threshold values. Let $Q := Q_1 \times \cdots \times Q_n$, where $Q_i := \{0, 1, \ldots, p_i\}$, $i \in \{1, \ldots, n\}$. Define the mapping $d^{\mathcal{A}} : \mathcal{D}_r \to Q$, where*

$$d_i^{\mathcal{A}}(D) := \begin{cases} 0 & \text{if } D_i = \{x \in \mathbb{R} \mid 0 \leq x < \theta_i^1\}, \\ q & \text{if } D_i = \{x \in \mathbb{R} \mid \theta_i^q < x < \theta_i^{q+1}\}, \\ p_i & \text{if } D_i = \{x \in \mathbb{R} \mid \theta_i^{p_i} < x \leq max_i\}. \end{cases}$$

Note that $d^{\mathcal{A}}$ is bijective.

To construct an update function on the discretized state space $Q$ that shares the dynamical properties of the PADE system, we naturally exploit the localization of the focal points in the discrete state space. Note that in general such a focal point may lie on a threshold plane, which by definition has no corresponding value in $Q$. As in the previous section, we exclude the comparatively small set of PADE systems with focal points on a threshold plane.

**Definition 4.2.** *Let $\mathcal{A}$ be a set of PADEs as in (2) such that all focal points lie in regulatory domains and let $d = d^{\mathcal{A}}$ be the mapping defined above. Define an update function $f^{\mathcal{A}} : Q \to Q$ with*

$$q \mapsto d(D_{\phi((d^{-1}(q))}),$$

*where $D_{\phi(D')}$ denotes the regulatory domain containing the focal point $\phi(D')$ of the regulatory domain $D'$.*

The update function defined above is uniquely determined by the threshold order and the position of the focal points relative to the thresholds, as described in the ordering constraints in the previous section. Consequently, the set of PADE systems satisfying given ordering constraints can be associated with a single discrete update function.

Before we take a close look at the dynamical properties conserved during the transformation from a given PADE system to a discrete update function, we shortly describe the reverse procedure associating a PADE system with a given update function.

**Definition 4.3.** *Let $f : Q \to Q$ be an update function. We denote by $PADE(f)$ the system of PADEs on $\Omega := \prod_{i=1}^{n}[0, max_i]$, $max_i \in \mathbb{R}_{>0}$ for all $i \in \{1, \ldots, n\}$, of the form $\dot{x}_i = F_i(x) - x_i$, $i \in \{1, \ldots, n\}$, where*

$$F_i(x) = \sum_{q \in Q} f_i(q) \prod_{j=1}^{n} S(x_j, q_j)$$

*with*

$$S(x_j, q_j) = \begin{cases} S^+(x_j, \theta_j^{q_j}) S^-(x_j, \theta_j^{q_j+1}) & \text{if } q_j \in \{1, \ldots, p_j - 1\}, \\ S^-(x_j, \theta_j^{q_1}) & \text{if } q_j = 0, \\ S^+(x_j, \theta_j^{q_{p_j}}) & \text{if } q_j = p_j, \end{cases}$$

*and $\theta_j^k = k - \frac{1}{2}$ for $j \in \{1, \ldots, n\}$, $k \in \{1, \ldots, p_j\}$.*

The choice of threshold values is generic, ensuring an obvious correspondence between the values $0, 1, \ldots, p_i$ in $Q_i$, $i \in \{1, \ldots, n\}$, and the intervals $[0, \theta_i^1]$, $(\theta_i^k, \theta_i^{k+1})$ for $k \in \{1, \ldots, p_i - 1\}$, and $(\theta_i^{p_i}, max_i]$. If we calculate the regulatory domains according to the threshold values and denote by $D_x$ the regulatory domain containing $x \in \mathbb{R}^n_{\geq 0}$, then we have $F_i(x) = f_i(d(D_x))$, where $d$ again denotes the corresponding discretization function. The focal point of domain $D$ is then given by $\phi(D) = F(x) = f(d(D))$ for all $x \in D$, or equivalently $\phi(d^{-1}(q)) \in d^{-1}(f(q))$ for $q \in Q$. We immediately see that all focal points of $PADE(f)$ lie in regulatory domains, thus we can apply Def. 4.2 to $PADE(f)$. By construction we then have $f^{PADE(f)} = f$.

In contrast, we generally do not have equality of PADE systems $\mathcal{A}$ and $PADE(f^{\mathcal{A}})$ due to the normalized form of $PADE(f^{\mathcal{A}})$. However, threshold order and relative focal point positions obviously coincide, i.e., $\mathcal{A}$ and $PADE(f^{\mathcal{A}})$ satisfy the same ordering constraints. In consequence, the two corresponding qualitative transition graphs are isomorphic, and only differ in the specific set of real vectors contained in corresponding domains, i.e., the designation of the vertices of the QTGs.

We summarize the preceding observations in the following proposition.

**Proposition 4.4.** *Let $f : Q \to Q$ be an update function and $\mathcal{A}$ be a PADE such that all focal points lie in regulatory domains. We then have*

$$f^{PADE(f)} = f \text{ and therefore } STG(f^{PADE(f)}) = STG(f).$$

*and*

$$QTG(\mathcal{A}) \cong QTG(PADE(f^{\mathcal{A}})),$$

*where '$\cong$' means isomorph.*

We again illustrate the content of this section using a simple example.

**Example 4.5.** *Our update function $f$ from Ex. 2.2 generates PADE($f$) with*

$$\begin{aligned} \dot{x}_1 &= 1 - S^+(x_1, \frac{1}{2})S^-(x_2, \frac{1}{2}) - x_1, \\ \dot{x}_2 &= 1 - S^+(x_1, \frac{1}{2})S^-(x_2, \frac{3}{2}) + S^-(x_1, \frac{1}{2})S^-(x_2, \frac{3}{2}) - x_2. \end{aligned}$$

*The inequalities (5) and (6) are satisfied, if we choose the parameters of the PADE system $\mathcal{A}$ introduced in Ex. 3.9 as $\alpha_1 = \alpha_2 = \gamma_2 = \lambda_1 = \lambda_2 = 1$, $\beta_1 = \beta_2 = -1$, $\theta_1^1, \theta_2^1 = \frac{1}{2}$ and $\theta_2^2 = \frac{3}{2}$. Therefore $PADE(f)$ belongs to the class of PADEs that are represented by $\mathcal{A}$.*

*Similarly, if we discretize $\mathcal{A}$ as described in Def. 4.2, we obtain the update function $f$ from Ex. 2.2, that is, $f^{\mathcal{A}} = f$.*

We have seen that we can associate a class of PADE systems characterized by the ordering constraints with a unique discrete update function. Furthermore, Proposition 4.4 shows that the information necessary for constructing a state transition resp. a qualitative transition graph is inherent in both representations. In that sense we can identify a class of PADE systems with the corresponding discrete update function. In the following we analyze differences and similarities of state and qualitative transition graph of a discrete function and the corresponding set of PADE systems, respectively.

# 5 Comparing the dynamics

Throughout this section we consider an update function $f : Q \to Q$ and the PADE system $\mathcal{A} := PADE(f)$ representing the class of PADE systems corresponding to $f$ according to the preceding section. In particular, discretization of $\mathcal{A}$ via the function $d := d^{\mathcal{A}}$ yields $f$. The aim of this section is the comparison of $STG(f) = (Q, E)$ and $QTG(\mathcal{A}) = (\mathcal{D}, \mathcal{T})$.

We start by analyzing whether the information encoded in $STG(f)$ is sufficient to derive $QTG(\mathcal{A})$ and vice versa. In the multi-valued other than in the Boolean setting, the state transition graph $STG(f)$ generally does not carry enough information to reconstruct $f$. However, by definition it is possible to derive the unitary update function $\tilde{f} := f^{STG(f)}$. We have already seen that $\tilde{f}(q) - q_j = sgn(f_j(q) - q_j)$ for all $q \in Q$, $j \in \{1, \ldots, n\}$. Furthermore, we know from the preceding section that $\phi(d^{-1}(q)) \in d^{-1}(f(q))$. Applying the definition for the relative position vector $v_j(d^{-1}(q), \phi(d^{-1}(q)))$ we immediately obtain the following lemma.

**Lemma 5.1.** *For all $q \in Q$ and $j \in \{1, \ldots, n\}$, we have*

$$v_j(d^{-1}(q), \phi(d^{-1}(q))) = \tilde{f}_j(q) - q_j, \tag{7}$$

*and similarly*

$$v_j(D, \phi(D)) = \tilde{f}_j(d(D)) - d_j(D)$$

*for all $D \in \mathcal{D}_r$.*

Given the right hand side of (7), we are able to reconstruct $STG(f)$ by definition of the state transition graph, while Lemma 3.8 ensures that we can build $QTG(\mathcal{A})$ knowing the relative position vectors given by the left hand side of the equation. In addition, given $STG(f)$ and $QTG(\mathcal{A})$, we can extract the unitary update function and the relative position vectors for regulatory domains. Consequently, we can state that we can construct $STG(f)$ given $QTG(\mathcal{A})$ and vice versa.

In the following we will see that despite this correspondence of $STG(f)$ and $QTG(\mathcal{A})$ it is difficult to relate the dynamical behaviors that the different graphs represent.

## 5.1 Edges

Comparing $STG(f)$ and $QTG(\mathcal{A})$ immediately yields a fundamental difference. The vertices of $STG(f)$ correspond to the regulatory domain vertices of $QTG(\mathcal{A})$, however, there is no representation of switching domains in the purely discrete setting. To answer the question whether $QTG(\mathcal{A})$ is just a finer representation of the dynamics encoded in $STG(f)$, we have to take a closer look at the correspondence of edges, and more generally paths, in both graphs. Based on the treatment of switching domains described in Sect. 3, we introduce a function associating with each switching domain the discretized values of its adjacent regulatory domains. We define the mapping $H : \mathcal{D} \to 2^Q$ by

$$H(D) := \begin{cases} \{d(D)\}, & \text{if } D \in \mathcal{D}_r, \\ \{d(D') \in Q \mid D' \in R(D)\} & \text{if } D \in \mathcal{D}_s, \end{cases}$$

where $2^Q$ is the power set of $Q$. For example, for a switching domain $D'$ of order one, $H(D')$ constitutes the set $\{d(D), d(\tilde{D})\}$ for the two regulatory domains $D$, $\tilde{D}$ adjacent to $D'$. Furthermore, we define for $i \in \{1, \ldots, n\}$

$$H_i(D) := \{q_i \in Q_i \mid q \in H(D)\}.$$

The relation between edges in $STG(f)$ and $QTG(\mathcal{A})$ turns out to be easily described if we focus on state transitions between regulatory domains and adjacent switching domains of order one.

**Proposition 5.2.** *Let $D \in \mathcal{D}_r$, and let $D' \subset \partial D$ be a switching domain of order one. Set $q := d(D)$ and denote by $q'$ the unique element in the set $H(D') \setminus H(D)$.*

*Then $(D, D') \in \mathcal{T}$ if and only if $(q, q') \in E$, and $(D', D) \in \mathcal{T}$ if and only if $(q, q') \notin E$.*

The statement implies that edges between two nodes $q, q' \in Q$, $q \neq q'$, in the STG always correspond to edges in the QTG for the switching domain $D'$ of order 1 with $H(D') = \{q, q'\}$, which agrees with observations of Chaves et al. for boolean discrete models [2], Richard et al. [9] and Gouzé and Sari [8]. It is a special case of Prop. 5.4 concerning regulatory domains and adjacent switching domains of arbitrary order.

To describe all outgoing edges of $q$, we still need to cover the case of $(q, q) \in E$, i.e., $q$ is a fixed point of $f$ as well as of $\tilde{f}$. We can immediately deduce from Lemma 5.1 that $v(D, \phi(D)) = 0$ for the regulatory domain $D = d^{-1}(q)$, if $q$ is a fixed point. Obviously, the reverse is also true. This observation proves the following proposition.

**Proposition 5.3.** *Let $D \in \mathcal{D}_r$, and let $q := d(D)$. Then $D$ has no outgoing edge in $QTG(\mathcal{A})$ if and only if $(q, q) \in E$.*

Now we can identify all edges in $STG(f)$ with specific edges in $QTG(\mathcal{A})$, or in the special case of loops, with the lack of outgoing edges of regulatory components. In turn, we have not yet understood how to describe all edges in $QTG(\mathcal{A})$ in terms of edges in $STG(f)$. A slight generalization of Prop. 5.2 leads to the following statement.

**Proposition 5.4.** *Let $D \in \mathcal{D}_r$, and let $D' \subset \partial D$ be a switching domain of order $k$. Set $q := d(D)$ and let $H'$ be the subset of $H(D') \setminus H(D)$ of the $k$ elements differing from $q$ in only one component.*

*Then $(D, D') \in \mathcal{T}$ if and only if $(q, q') \in E$ for all $q' \in H'$, and $(D', D) \in \mathcal{T}$ if and only if $(q, q') \notin E$ for all $q' \in H'$.*

This is in turn a corollary of a more general statement we will prove below. The proposition, together with Prop. 5.3, shows that we can derive all outgoing edges of a vertex $q$ leading to a state $q' \neq q$ in $STG(f)$ from the outgoing edges of the corresponding regulatory domain $D = d^{-1}(q)$ in $QTG(\mathcal{A})$ and vice versa.

However, we encounter difficulties when considering the dynamics on switching domains. Lemma 5.1 ensures that we can derive the outgoing edges of a switching domain from the information inherent in $STG(f)$, but there is no clear edge correspondence to edges in $STG(f)$. The following theorem describes the correspondences. The proof is rather technical but basically consists of a translation of the conditions given in Def. 3.5 into a description using the local structure of $STG(f)$ via Lemma 5.1.

**Theorem 5.5.** *Let $D \in \mathcal{D}$ and $D' \subset \partial D$. Denote by $I$ the index set of switching variables in $D$ and $I'$ the index set of switching variables in $D'$. Then*

1. *$(D, D') \in \mathcal{T}$ if and only if*

   (a) *for all $i \in I$ exist $q^1, q^2 \in H(D)$, $q^1 \neq q^2$, such that $p_i^1 \leq q_i^1$, $q_i^2 \leq p_i^2$ for all $p^1, p^2 \in Q$ with $(q^1, p^1), (q^2, p^2) \in E$, and*

$$\exists\, l \in \{1,2\},\, p \in H(D):\ (q^l, p) \in E \ \wedge\ p_i \neq q_i^l, \qquad \text{if } q_i^1 = q_i^2,$$

*and*

$$\exists\, p^1, p^2 \in H(D):\ (q^1, p^1), (q^2, p^2) \in E \ \wedge\ p_i^1 < p_i^2, \qquad \text{if } q_i^1 > q_i^2,$$

*(b) for all $i \in I' \setminus I$ there exists $q \in H(D)$ and $q' \in H(D') \setminus H(D)$ with $q_i \neq q'_i$ and $(q, q') \in E$,*

*2. $(D', D) \in \mathcal{T}$ if and only if*

*(a) for all $i \in I$ exist $q^1, q^2 \in H(D)$, $q^1 \neq q^2$, such that $p_i^1 \leq q_i^1$, $q_i^2 \leq p_i^2$ for all $p^1, p^2 \in Q$ with $(q^1, p^1), (q^2, p^2) \in E$, and*

$$\exists\, l \in \{1,2\},\, p \in H(D):\ (q^l, p) \in E \ \wedge\ p_i \neq q_i^l, \qquad \text{if } q_i^1 = q_i^2,$$

*and*

$$\exists\, p^1, p^2 \in H(D):\ (q^1, p^1), (q^2, p^2) \in E \ \wedge\ p_i^1 < p_i^2, \qquad \text{if } q_i^1 > q_i^2,$$

*(b) for all $i \in I' \setminus I$ there exists $q \in H(D)$ and $q' \in H(D') \setminus H(D)$ such that $q_i \neq q'_i$, $q'_j = q_j$ for all $j \neq i$ and $(q, q') \notin E$.*

*Proof.* The respective two conditions for the existence of transitions in $QTG(A)$ are simply a reformulation of the two conditions given in Def. 3.5 in the context of edges of $STG(f)$.

We start by showing that the condition $V(D, \Psi(D)) \neq \emptyset$ is equivalent to condition 1. $(a)$ in the theorem. If $D$ is a regulatory domain, then $V(D, \Psi(D)) \neq \emptyset$ by definition and condition 1. $(a)$ is true by default since $I$ is empty. Let us now assume $D \in \mathcal{D}_s$. Figure 3 illustrates the conditions in $(a)$ ensuring $V(D, \Psi(D)) \neq \emptyset$.

Note that by definition of asynchronous successors and since $f$ is a function, we have for all $q \in Q$ and all $i \in \{1, \ldots, n\}$ that either $q_i \leq p_i$ for all $p \in AS(q)$ or $q_i \geq p_i$ for all $p \in AS(q)$, and $AS(q) \neq \emptyset$. Also, we observe that $|q_i^1 - q_i^2| \leq 1$ for all $q^1, q^2 \in H(D)$ and $i \in \{1, \ldots, n\}$ by definition of $H(D)$. In particular, $1 + \min_{\tilde{q} \in H(D)} \tilde{q}_i = \max_{\tilde{q} \in H(D)} \tilde{q}_i$ for all $i \in I$.

First, we transform the condition $V(D, \Psi(D)) \neq \emptyset$ into a condition expressed in terms of the unitary update function $\tilde{f}$. According to Prop. 3.7, $V(D, \Psi(D)) \neq \emptyset$ is equivalent to $\min_{\tilde{D} \in R(D)} v_i(D, \phi(\tilde{D})) = -1$ and $\max_{\tilde{D} \in R(D)} v_i(D, \phi(\tilde{D})) = 1$ for all $i \in I$. For fixed $i \in I$ and due to the adjacency of each $\tilde{D} \in R(D)$ to $D$, the condition concerning the minimum can be expressed as $\min_{\tilde{D} \in R(D)} v_i(\tilde{D}, \phi(\tilde{D})) \in \{-1, 0\}$, where the minimum can only be zero if it is assumed for a domain $\tilde{D}$ satisfying $w_i = -1$, where $V(D, \tilde{D}) = \{w\}$. Applying Lemma 5.1, this can be formulated similarly for $\min_{\tilde{D} \in R(D)} (\tilde{f}_i(d(\tilde{D})) - d_i(\tilde{D}))$. Using the definition of $H(D)$ and the observations concerning $H(D)$ above yields the equivalent condition $\min_{\tilde{q} \in H(D)} \tilde{f}_i(\tilde{q}) \leq \min_{\tilde{q} \in H(D)} \tilde{q}_i$. Analogously, we can derive a similar condition equivalent to $\max_{\tilde{D} \in R(D)} v_i(D, \phi(\tilde{D})) = 1$ and since $i$ is a switching variable of $D$, we obtain

$$\min_{\tilde{q} \in H(D)} \tilde{f}_i(\tilde{q}) \leq \min_{\tilde{q} \in H(D)} \tilde{q}_i < \max_{\tilde{q} \in H(D)} \tilde{q}_i \leq \max_{\tilde{q} \in H(D)} \tilde{f}_i(\tilde{q}). \tag{8}$$

We now show the equivalence of conditions $(a)$ and (8). Let us first assume that (8) holds. Choose $q^1, q^2 \in H(D)$ such that $\tilde{f}_i(q^1) = \min_{\tilde{q} \in H(D)} \tilde{f}_i(\tilde{q})$ and $\tilde{f}_i(q^2) = \max_{\tilde{q} \in H(D)} \tilde{f}_i(\tilde{q})$. Then,

according to the preliminary observations and (8), $p_i \leq q_i^1$ for all $p \in AS(q^1)$, and $q_i^2 \leq p_i$ for all $p \in AS(q^2)$. If $q_i^1 < q_i^2$, nothing is left to prove. If $q_i^1 > q_i^2$, then $q_i^1 = \max_{\tilde{q} \in H(D)} \tilde{q}_i$ and $q_i^2 = \min_{\tilde{q} \in H(D)} \tilde{q}_i$. Condition $(a)$ follows from (8), the choice of $q^1$ and $q^2$ and the definition of asynchronous successor. Otherwise, let us assume that $q_i^1 = q_i^2 = \min_{\tilde{q} \in H(D)} \tilde{q}_i$. Then $q_i^2 < \max_{\tilde{q} \in H(D)} \tilde{f}_i(\tilde{q}) = \tilde{f}_i(q^2)$. By the definition of asynchronous update, we then find $p \in AS(q^2)$ with $p_i = q_i^2 + 1 = \max_{\tilde{q} \in H(D)} \tilde{q}_i$ and $p_j = q_j$ for all $j \neq i$. In particular, $p \in H(D)$, and thus condition $(a)$ holds. Analogous reasoning holds for $q_i^1 = q_i^2 = \max_{\tilde{q} \in H(D)} \tilde{q}_i$.

Let us now assume that condition $(a)$ of the theorem holds, and choose $q^1, q^2$ according to the statement. Then $\tilde{f}_i(q^1) \leq q_i^1$ and $\tilde{f}_i(q^2) \geq q_i^2$. By definition of $H(D)$, we always have $\min_{\tilde{q} \in H(D)} \tilde{q}_i < \max_{\tilde{q} \in H(D)} \tilde{q}_i$. If $q_i^1 < q_i^2$, then condition (8) follows immediately.

If $q_i^1 = q_i^2$, let us again reason for $q_i^1 = q_i^2 = \min_{\tilde{q} \in H(D)} \tilde{q}_i$, the other case being similar. Then $\min_{\tilde{q} \in H(D)} \tilde{f}_i(\tilde{q}) \leq \tilde{f}_i(q^1) \leq q_i^1 = \min_{\tilde{q} \in H(D)} \tilde{q}_i$. Now, choose $p \in H(D)$ according to $(a)$. Then $p$ has to be successor of $q^2$, since otherwise $p_i < q_i^1 = \min_{\tilde{q} \in H(D)} \tilde{q}_i$ in contradiction to $p \in H(D)$. It follows that $p_i = \tilde{f}_i(q^2)$, $\max_{\tilde{q} \in H(D)} \tilde{q}_i = p_i$ and $\min_{\tilde{q} \in H(D)} \tilde{q}_i = q_i^2 < \max_{\tilde{q} \in H(D)} \tilde{q}_i = p_i = \tilde{f}_i(q^2) \leq \max_{\tilde{q} \in H(D)} \tilde{f}_i(\tilde{q})$. Thus, we have (8).

In case $q_i^2 < q_i^1$, choose $p^1, p^2$ according to $(a)$. Then $q_i^2 + 1 = p_i^2$ and $p_i^2 = p_i^1 + 1$, and thus $q_i^2 = p_i^1$. It follows that $\min_{\tilde{q} \in H(D)} \tilde{f}_i(\tilde{q}) \leq \tilde{f}_i(q^1) \leq p_i^1 = q_i^2 = \min_{\tilde{q} \in H(D)} \tilde{q}_i$. Similar reasoning for $q^1$ yields $q_i^1 = \max_{\tilde{q} \in H(D)} \tilde{q}_i \leq \max_{\tilde{q} \in H(D)} \tilde{f}_i(\tilde{q})$, and in summary we obtain (8).

We have shown that the condition 1.$(a)$ (and 2.$(a)$) of the theorem are equivalent to the conditions 1.$(a)$ (and 2.$(a)$) of Def. 3.5. Now, we show the equivalence of the corresponding $(b)$ conditions, provided $(a)$, and thus $V(D, \Psi(D)) \neq \emptyset$, holds. Condition 1.$(b)$ of Def. 3.5 states that there exists $\nu \in V(D, \Psi(D))$ such that $\nu_i w_i = 1$ for every $i \in I' \setminus I$, where $V(D, D') = \{w\}$.

Let us first remark that $w_i \neq 0$ if and only if $i \in I' \setminus I$. Moreover, $w_i = q_i' - q_i$ for all $q \in H(D)$, $q' \in H(D') \setminus H(D)$ for $i \in I' \setminus I$.

Now, let $i \in I' \setminus I$, i.e., $w_i \neq 0$, and choose $q \in H(D)$, $q' \in H(D') \setminus H(D)$ with $(q, q') \in E$ and $q_i \neq q_i'$ according to condition 1.$(b)$ of the theorem. Let us assume that $w_i = 1$, the case $w_i = -1$ can be treated analogously. Then $1 = w_i = q_i' - q_i$, i.e., $q_i' > q_i$. It follows that $\tilde{f}_i(q) > q_i$, and thus $\max_{\tilde{q} \in H(D)} \tilde{f}_i(\tilde{q}) - \tilde{q}_i = 1$. If $D$ is a regulatory domain, then $v_i(D, \phi(D)) = 1$ by Lemma 5.1. If $D$ is a switching domain, then $1 \in [\min_{\tilde{q} \in H(D)} \tilde{f}_i(\tilde{q}) - \tilde{q}_i, \max_{\tilde{q} \in H(D)} \tilde{f}_i(\tilde{q}) - \tilde{q}_i] = V_i(D, \Psi(D))$ according to Prop. 3.7 and Lemma 5.1. In both cases, there exists $\nu_i \in V(D, \Psi(D))$ with $\nu_i w_i = 1$. Since the definition of $V(D, \Psi(D))$ for regulatory domains and Prop. 3.7 allow for a componentwise argument, the existence of a vector $\nu \in V(D, \Psi(D))$ with $\nu_i w_i = 1$ for all $i \in I' \setminus I$ follows.

To show the reverse statement, assume that there exists $\nu \in V(D, \Psi(D))$ with $\nu_i w_i = 1$ for all $i \in I' \setminus I$, and choose $i \in I' \setminus I$. Again we restrict ourselves to the exemplary case $w_i = 1$. Then $\nu_i = 1$, and therefore there exists $q \in H(D)$ with $\tilde{f}_i(q) - q_i = 1$ according to Lemma 5.1 and Prop. 3.7. In particular, $(q, q') \in E$ for $q' \in Q$ with $q_i' = q_i + 1$ and $q_j' = q_j$ for all $j \neq i$. Then $q' \notin H(D)$, since $q_i' \neq q_i$ and $i \notin I$, but $q' \in H(D')$, since $q_i' = q_i + w_i$. Thus, condition 1.$(b)$ of the theorem holds.

Lastly, we show equivalence of the conditions 2.$(b)$ of the theorem and Def. 3.5. Note that other than in the notation of Def. 3.5 2.$(b)$, we have $D' \subset \partial D$. Thus, now let $w$ be the single element of $V(D', D)$.

Suppose there exists $\nu \in V(D, \Psi(D))$ with $\nu_i w_i \neq -1$ for all $i \in I' \setminus I$. Let $i \in I' \setminus I$, then $w_i \neq 0$. Again, we only show the exemplary proof for the case $w_i = 1$. Then $\nu_i \neq -1$, and thus $\max_{\tilde{q} \in H(D)} \tilde{f}_i(\tilde{q}) - \tilde{q}_i \geq 0$ according to Prop. 3.7 and Lemma 5.1. Then, there exists

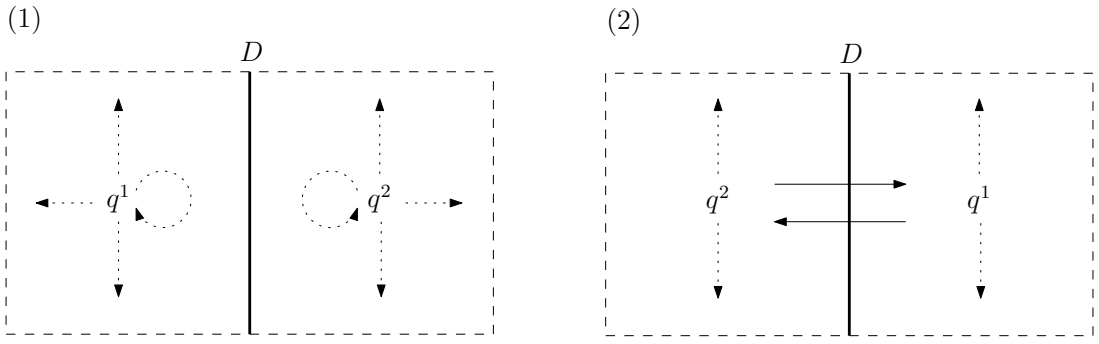



Figure 3: Illustration of condition 1.($a$) of Theorem 5.5 for $D$ being a switching domain of order 1 depicted as solid vertical line between two regulatory domains. The property $V(D, \Psi(D)) \neq \emptyset$ translates to edge constraints for outgoing edges of discrete states $q \in H(D)$. On the left, permissable edges for the case $q_i^1 < q_i^2$. On the right, permissable edges for $q_i^2 < q_i^1$ with the mandatory edges depicted as solid lines.

$q \in H(D)$ with $q_i \leq \tilde{f}_i(q)$, which yields $q_i \leq p_i$ for all asynchronous successors $p \in AS(q)$. Now, let $q' \in H(D') \setminus H(D)$ with $q'_i \neq q_i$ and $q'_j = q_j$ for all $j \neq i$. Since $w_i = 1$, we have $q'_i < q_i$. In particular, $q'$ cannot be an asynchronous successor of $q$, i.e., $(q, q') \neq E$.

In turn, given $i \in I' \setminus I$ and $q, q'$ according to condition 2.($b$), i.e., $q \in H(D)$, $q' \in H(D') \setminus H(D)$ with $q_i \neq q'_i$, $q_j = q'_j$ for all $j \neq i$, and $(q, q') \notin E$, then $w_i = q_i - q'_i$. Again, let us just focus on the case $w_i = 1$, i.e., $q_i > q'_i$. Since $(q, q') \notin E$, we have $p_i \geq q_i$ for all $p \in AS(q)$, i.e., $\tilde{f}_i(q) - q_i \in \{0, 1\}$. It follows from Prop. 3.7 and Lemma 5.1 that $\max_{\tilde{D} \in R(D)} v_i(\tilde{D}, \phi(\tilde{D})) \geq 0$. Thus we can find $\nu \in V(D, \Psi(D))$ with $\nu_i \neq -1$, and, in particular, $\nu_i w_i \neq -1$. Again, the definition of $V(D, \Psi(D))$ for regulatory domains and Prop. 3.7 allow for a componentwise argument, and we can fulfill condition 2.($b$) of Def. 3.5. □

The above theorem completely describes the correspondences between edges in $QTG(\mathcal{A})$ and $STG(f)$, and allows to construct one from the other based on the graph topology. It can be seen as a tool for elucidating the correspondences between more complex structures, such as paths or attractors, in the two graphs. On the one hand, it can be used for proofs building on local considerations concerning the edges involved, on the other hand, it provides ideas for the construction of counterexamples. In the following section, we illustrate both uses of the theorem.

## 5.2 Paths and attractors

We have seen that comparing edges of $STG(f)$ and $QTG(\mathcal{A})$ is not always straightforward. Clearly, this difficulty carries over to comparisons of paths in the two graphs. However, in easy cases we can find conditions ensuring the existence of corresponding paths. The following statement simply applies Prop. 5.2 repeatedly to obtain a result for paths.

**Proposition 5.6.** *There exists a path $(D^1, \ldots, D^{2k+1})$ in $QTG(\mathcal{A})$ with $D^i \in \mathcal{D}_r$ for $i \in \{1, \ldots, 2k+1\}$ odd and $D^i$ a switching domain of order 1 for $i \in \{1, \ldots, 2k+1\}$ even, if*

*and only if* $(q^0, q^1, \ldots, q^k)$ *is a path in* $STG(f)$ *such that* $(q^j, q^{j-1}) \notin E$ *for all* $j \in \{1, \ldots, k\}$ *and* $q^i = d(D^{2i+1})$ *for all* $i \in \{0, \ldots, k\}$.

The correspondence between paths may be lost when transitions between switching domains are included. In our running example, e.g., we can find a path from state (1,0) to (0,1) via (0,0) in the state transition graph. In the corresponding QTG, there is no path visiting the respective regulatory domains. In fact, the regulatory domain corresponding to (0,1) is not reachable from the regulatory domain corresponding to (1,0), as can be seen in Fig. 4 (a).

In turn, it is easy to construct an example showing that a path in the QTG does not necessarily correspond to a comparable path in the STG. One such example is shown in Fig. 4 (b). Let us consider a Boolean STG, i.e., an STG with four vertices and a corresponding QTG with four regulatory domains. We connect the regulatory domains corresponding to (0,0) and (1,1) with a path via the central switching domain of order 2. Then, Theorem 5.5, or Prop. 5.4, yields the conditions for the corresponding state transition graph. There have to be edges from (0,0) to (1,0) and to (0,1), and there are no edges allowed from (1,1) to (1,0) or to (0,1). We can now complete the description of the system by adding edges to the STG such that there are no edges leading from (0,1) or (1,0) to (1,1). Thus, we obtain a system where (1,1) is not reachable from (0,0) in the STG while the reachability property holds for the corresponding regulatory domains in the QTG.

The two examples allow us to phrase the following statement.

**Proposition 5.7.** *Given a path* $(D^1, \cdots, D^k)$ *in* $QTG(\mathcal{A})$ *with* $D^1, D^k \in \mathcal{D}_r$ *including at least one switching domain of order greater than one, there does not necessarily exist a path from* $d(D^1)$ *to* $d(D^k)$ *in* $STG(f)$. *In turn, if a path* $(q^1, \cdots, q^k)$ *in* $STG(f)$ *includes at least one* $j = 2, \cdots, k$ *such that* $(q^j, q^{j-1}) \in E$, *then there does not necessarily exist a path from* $d^{-1}(q^1)$ *to* $d^{-1}(q^k)$ *in* $QTG(\mathcal{A})$.

The proposition establishes that reachability properties are not conserved between the two graphs, at least, if we strictly identify the vertices in $STG(f)$ with the regulatory domains in $QTG(\mathcal{A})$. Even considering a weaker reachability correspondence where we identify a discrete state with the respective regulatory domain and the switching domains in its boundary, does not yield matching properties, as can again be seen in our running example in Fig. 4 (a). In the state transition graph, we can find a path from (1,0) to (0,2), but no path in the QTG starting in the regulatory domain corresponding to (1,0) or in any of its adjacent switching domains reaches the regulatory domain corresponding to (0,2) or one of its adjacent switching domains.

A further important characteristic of $QTG(\mathcal{A})$ and $STG(f)$ are their respective attractors. The next definition introduces our terminology.

**Definition 5.8.** *Let* $G$ *be a directed graph and* $S$ *a subset of the nodes of* $G$. *The set* $S$ *is* strongly connected *if any two nodes in* $S$ *are connected by a path in* $S$. *The set* $S$ *is a* trap set *if there is no path leaving* $S$. *An* attractor *of a graph is a strongly connected trap set. A* steady state *is an attractor consisting of a single node. A* simple cycle *is an attractor of cardinality greater than one where every node has only one outgoing edge. A* complex attractor *is an attractor that has at least one node which has two outgoing edges.*

Note that we consider each node set of cardinality one to be strongly connected by default, i.e., not depending of the existence of a loop on the respective node. The steady states in a discrete state transition graph correspond to the fixed points of the update function $f$, and by

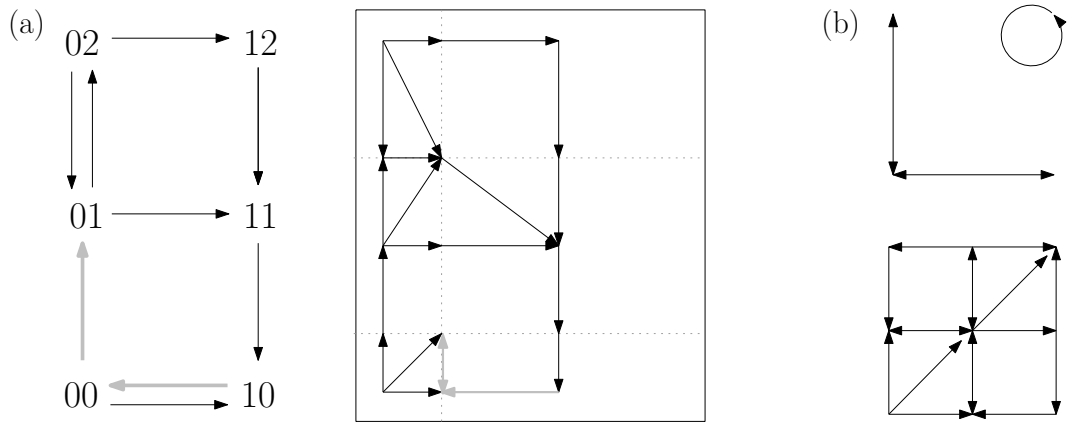



Figure 4: In (a), STG and QTG of the running example. Gray edges indicate a path from (1,0) to (0,1) in the STG on the left, and the only path starting in the regulatory domain corresponding to (1,0) in the QTG on the right. In (b), the STG of a two component Boolean network with the corresponding QTG below (vertices are not explicitly depicted).

definition there exists an edge $(q, q)$ for each fixed point $q$. In contrast, the steady states of the QTG are, by definition of the transitions, nodes without outgoing edges. Note that here a steady state is simply a singleton terminal strongly connected component in a graph. There is no implicit statement about the stability of such a steady state included, i.e., it does not necessarily correspond to an asymptotically stable steady state in a PADE system (see e.g. [10]).

Attractors of cardinality greater than one are often called cyclic. Both complex attractors and simple cycles are then cyclic attractors. We often denote a simple cycle $\{s^1, \ldots, s^k\}$ by the path $(s^1, \ldots, s^k, s^1)$ traversing the cycle. Let us first make an observation concerning simple cycles in qualitative transition graphs.

**Lemma 5.9.** *If there exists a simple cycle in $QTG(\mathcal{A})$ which includes a regulatory domain, then the cycle alternates between regulatory domains and switching domains of order one.*

*Proof.* Let us start with a general observation. The first item of Def. 3.5 (or of Theorem 5.5) immediately yields that whenever there is a transition from a domain $D$ of order $k$, where we set $k = 0$ if $D$ is a regulatory domain, to a domain $D''$ of order at least $k + 2$, then we can find a domain $D'$ of order $k + 1$ with $(D, D') \in \mathcal{T}$.

Now, let us assume $D$ is a regulatory domain included in a simple cycle $\mathcal{C}$. Then $D$ has only one outgoing edge in QTG and therefore its successor $D'$ in $\mathcal{C}$ has to be a switching domain of order 1. Let $i$ be the switching component of $D'$. We have $v_j(D, \phi(D)) = 0$ for all $j \neq i$. Suppose now that the successor $D''$ of $D'$ in $\mathcal{C}$ is not a regulatory domain. Then $D''$ is a switching domain of order greater than one adjacent to $D$. Since $(D', D'') \in \mathcal{T}$, we have $V(D', \Psi(D')) \neq \emptyset$. Furthermore, by Prop. 3.7, we have $V_i(D', \Psi(D')) = \{0\}$ and, since $D \in R(D')$, $0 \in V_j(D'\Psi(D'))$ for $j \neq i$. Then $(D'', D') \in \mathcal{T}$ by Def. 3.5. Since $D$ is in $\mathcal{C}$, there has to be a path from $D'$ to $D$ in $\mathcal{C}$. Consequently, there has to be an edge leaving the cycle $(D', D'', D')$. This is a contradiction, since $D'$ and $D''$ are in $\mathcal{C}$ and only have one outgoing edge each. □

Let us now return to the comparison of $QTG(\mathcal{A})$ and $STG(f)$. The following proposition

captures some easy relations between attractors in the respective graphs.

**Proposition 5.10.** *The following relations hold for attractors in $QTG(\mathcal{A})$ and $STG(f)$.*

1. *A regulatory domain $D$ is a steady state in $QTG(\mathcal{A})$ if and only if $d(D)$ is a steady state in $STG(f)$. A switching domain $D$ of order 1 is a steady state in $QTG(\mathcal{A})$ if and only if $H(D)$ is a simple cycle in $STG(f)$.*

2. *There is a simple cycle including a regulatory domain in $QTG(\mathcal{A})$, if and only if there is a simple cycle $(D^1, D^2, \ldots, D^{2m}, D^1)$ with $D^{2j} \in \mathcal{D}_r$ for $j \in \{1, \ldots, m\}$ in $QTG(\mathcal{A})$, if and only if $(d(D^2), d(D^4), \ldots, d(D^{2m}), d(D^2))$ is a simple cycle in $STG(f)$.*

*Proof.* 1. The first statement immediately follows from Prop. 5.3. Now, let $\{q, q'\}$ be a simple cycle in $STG(f)$ and $D$ the switching domain of order one with $H(D) = \{q, q'\}$. According to Theorem 5.5, if there existed an edge from $D$ to a switching domain $D'$ of higher order, then we could find $p \in H(D)$, $p' \in H(D') \setminus H(D)$, i.e., $p \in \{q, q'\}$ and $p' \notin \{q, q'\}$, with $(p, p') \in E$, which would be contradictory to $H(D)$ being a simple cycle. If there existed an edge from $D$ to a regulatory domain $\tilde{D}$, then, since $H(D) \setminus H(\tilde{D}) = \{q^1\}$ and $H(\tilde{D}) = \{q^2\}$ for some $q^1, q^2 \in H(D)$ with $q^1 \neq q^2$, we would have $(q^2, q^1) \notin E$ according to Theorem 5.5, which contradicts $\{q, q'\}$ being a simple cycle. In summary, $D$ has no outgoing edges and is a steady state.

If $H(D) = \{q, q'\}$ is not a simple cycle, then either one of the edges $(q, q')$, $(q', q)$ is missing in $STG(f)$ or there exists an edge leaving $H(D)$. In the first case, there exists a transition from $D$ to a regulatory domain, as we can see immediately from Theorem 5.5. If $(q, q'), (q', q) \in E$, then condition 1.($a$) of Theorem 5.5 holds for $D$. If we find an additional edge leaving $H(D)$, then condition 1.($b$) of the Theorem holds as well, and we find an edge from $D$ to some switching domain. In any case $D$ is not a steady state.

2. Lemma 5.9 ensures that a simple cycle containing a regulatory domain already has the specific shape $(D^1, D^2, \ldots, D^{2m}, D^1)$ with $D^{2j} \in \mathcal{D}_r$ for $j \in \{1, \ldots, m\}$. The reverse is obviously true. If $\mathcal{C}$ is such a simple cycle in $QTG(\mathcal{A})$, then $(d(D^2), d(D^4), \ldots, d(D^{2m}), d(D^2))$ is a simple cycle in $STG(f)$ according to Prop. 5.2. If, on the other hand, $(d(D^2), d(D^4), \ldots, d(D^{2m}), d(D^2))$ is a simple cycle in $STG(f)$, then, again according to Prop. 5.2, each regulatory domain $D^{2j}$ has only one outgoing edge, namely $(D^{2j}, D^{2j+1})$, where $2m + 1$ is identified with index 1. For the switching domains $D^{2j+1}$ in the cycle, we can derive the existence of only one outgoing edge, namely $(D^{2j+1}, D^{2j+2})$, from Theorem 5.5 1.($a$). □

The above proposition only captures very basic correspondences. The situation becomes much less clear if we consider attractors in the QTG containing switching domains of order greater than one, as we illustrate in the following.

Consider again the STG and the QTG of the running example as given in Fig. 4 (a). We can find a simple cycle in the QTG between the switching domain of order 2 corresponding to the smallest thresholds for both components and the switching domain of order 1 below it. However, in the STG there exists only one attractor, which is a complex attractor consisting of the entire state space.

In a complex attractor in the STG, we can always find a node $q$ that has more than one outgoing edge. It immediately follows from Prop. 5.4 that there exists a transition from the corresponding regulatory domain $D$ to a switching domain of order greater than one in the QTG. The second statement of Theorem 5.5 yields conditions for transitions leaving switching domains that potentially produce reachability properties in the QTG not matching those of

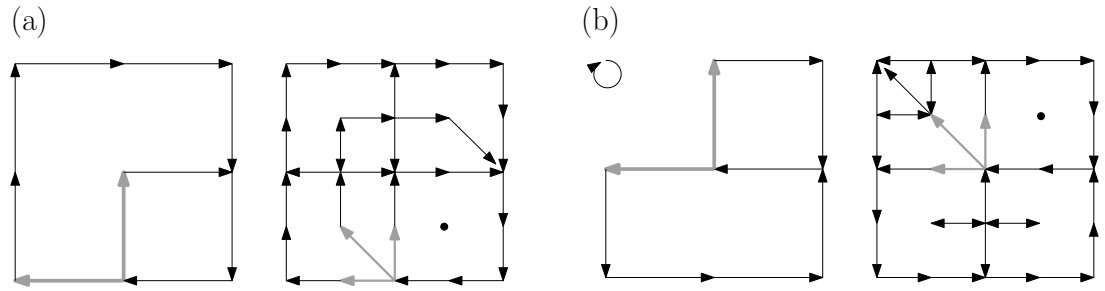



Figure 5: Two examples for networks with two components and three activity levels for each component. In each case, the STG is depicted on the left, the corresponding QTG on the right. Multiple outgoing edges of a node in the STG and the outgoing edges of the corresponding node in the QTG are shown in gray.

the STG. In particular, we can use them to merge attractors, so to speak. This situation is illustrated in Fig. 5 for an example network consisting of two components with three activity levels each. In (a), we see a system for which the STG consists of a single complex attractor. All regulatory domains in the corresponding QTG also belong to a complex attractor. In (b), the STG contains a steady state and a complex attractor. There is only one state with two outgoing edges in the STG. Those edges generate an edge from the regulatory domain in the center of the corresponding QTG to the switching domain $D''$ of order 2 representing the state consisting of the lower threshold level of the first and the higher of the second component. By constructing the state (0,2) as a steady state, we generate edges in the QTG that lead from $D''$ to the regulatory state corresponding to (0,2). In consequence, the QTG has no complex attractor. Let us lastly consider a switching domain $D$ of order greater than one which is a steady state in the QTG. Translating the condition of $D$ having no outgoing edges obviously imposes constraints on a corresponding STG via Theorem 5.5. However, these constraints are generally not strong enough to link $D$ to some unique structure in the STG. In Fig. 6, we see two examples, where both QTGs contain a switching domain of order 2, which is a steady state. However, the corresponding STGs differ, the one in (a) consisting of a complex attractor and the one in (b) consisting of a simple cycle.

We have seen that the relations between attractors in corresponding STGs and QTGs is not clear-cut, the examples illustrating that, in general, neither number nor character of attractors are preserved. Further work exploiting Theorem 5.5 could lead to useful characterizations of systems generating corresponding behaviors and generally elucidate the network properties destroying such correspondences. We close this section with a last general observation linking properties of the STG and the QTG.

**Proposition 5.11.** *Let $\mathcal{R} := R_1 \times \cdots \times R_n \subset Q$ be a discrete hyperrectangle, i.e., $R_i$ is an integer interval $[a_i, b_i] \subset Q_i$ for all $i \in \{1, \ldots, n\}$. Then $\mathcal{R}$ is a trap set in $STG(f)$ if and only if $U := \{D \mid H(D) \subset \mathcal{R}\}$ is a trap set in $QTG(\mathcal{A})$.*

*Proof.* Let $U$ be a trap set. Let $q \in \mathcal{R}$ and $q' \in Q$ with $(q, q') \in E$. Then Prop. 5.2 ensures that there exists $D' \subset \partial D$, $D := d^{-1}(q)$, with $(D, D') \in \mathcal{T}$ and $H(D') = \{q, q'\}$. Since $U$ is a trap set, we have $D' \in U$. Then, by definition, $H(D') \in \mathcal{R}$, i.e., $q' \in \mathcal{R}$.

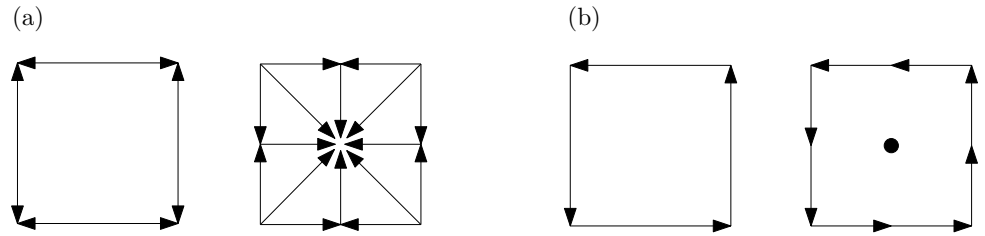



Figure 6: STGs and QTGs for networks with two components and two activity levels for each component, the STG shown on the left, the corresponding QTG on the right of each figure. Both the QTGs in (a) and (b) have a steady state in the switching domain of order two.

Now, let $\mathcal{R}$ be a trap set. Let $D \in U$ and $D' \in \mathcal{D}$ with $(D, D') \in \mathcal{T}$. If $D \subset \partial D'$, then $H(D') \subset H(D) \subset \mathcal{R}$, and therefore, by definition, $D' \in U$. If $D' \subset \partial D$, Theorem 5.5, 1.($b$) yields the existence of an edge $(q, q') \in E$ such that $q \in H(D)$, i.e., $q \in \mathcal{R}$, and $q' \in H(D') \setminus H(D)$ with $q_i \neq q'_i$ for all indices $i$ indicating switching variables of $D'$ but not of $D$. Since $\mathcal{R}$ is a hyperrectangle, we then have $H(D') \subset \mathcal{R}$, that is, $D' \in U$. □

The above proposition may be helpful in elucidating the correspondences of attractors in the STG and the QTG further. Since a trap set always contains at least one attractor, we can relate attractors that we can separate using hyperrectangles

## 6 Discussion and perspectives

We have seen that the Thomas formalism and the qualitative representation of PADE systems are closely related. In particular, we can identify each class of PADE systems characterized by the ordering constraints on threshold values and focal points with a discrete update function. In fact, the information inherent in the state transition graph of the update function is sufficient to derive the qualitative transition graph of the corresponding PADE system and vice versa. We have further clarified the relation between the two graphs by characterizing each edge in the QTG via conditions pertaining edges in the STG. Locally, the topological correspondences between the two graphs are thus completely described.

Building upon the local conditions, we state a number of observations relating more complex structures in the two graphs. Although some properties involving only regulatory domains and switching domains of order one are preserved, we see that the involvement of switching domains of greater order often destroys correspondences between the two graphs. In particular, essential features such as reachability properties or attractor characteristics do not generally coincide. Consequently, the QTG cannot be interpreted as a straightforward refinement of a Thomas model.

Motivated by our findings, there are several questions we want to focus on in future work. Firstly, we want to better understand and characterize the network properties that lead to substantial differences, e.g., in the number of attractors, in the dynamics of Thomas and the PADE models. Secondly, we want to extend the analysis to closely related formalisms. The qualitative representation of PADE systems has been further refined in recent years, capturing the behavior within regulatory domains more precisely [1]. Also, there exist approaches that

allow the integration of threshold values directly into the Thomas formalism [13, 9]. Clarifying the relation between the different approaches may allow, on the one hand, to transfer available results and analysis methods from one formalism to the other. On the other hand, progress in this direction may be helpful when deciding on the most suitable and efficient modeling framework in application.

## References

[1] G. Batt, H. de Jong, M. Page, and J. Geiselmann. Symbolic reachability analysis of genetic regulatory networks using discrete abstractions. *Automatica*, 44(4):982–989, 2008.

[2] M. Chaves, L. Tournier, and J.-L. Gouzé. Comparison between Boolean and piecewise affine differential models for genetic networks. *Acta Biotheoretica*, 58(2):217–232, 2010.

[3] H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselmann. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull. Math. Biol.*, 66(2):301–40, 2004.

[4] R. Edwards. Analysis of continuous-time switching networks. *Phys. D*, 146(1-4):165–199, 2000.

[5] A. Fillipov. *Differential equations with discontinuous righthand sides*. Springer, 1988.

[6] L. Glass and S. Kauffman. Co-operative components, spatial localization and oscillatory cellular dynamics. *J. Theoret. Biol.*, pages 219–237, 1972.

[7] L. Glass and S. A. Kauffman. The logical analysis of continuous, non-linear biochemical control networks. *J. Theoret. Biol.*, pages 103–129, 1973.

[8] J.-L. Gouzé and T. Sari. A class of piecewise linear differential equations arising in biological models. *Dyn. Syst.*, 17(4):299–316, 2002.

[9] A. Richard, G. Bernot, and J.-P. Comet. R. Thomas; Modeling of Biological Regulatory Networks: Introduction of Singular States in the Qualitative Dynamics. *Fund. Inform.*, 65:373–392, 2005.

[10] E. H. Snoussi. Qualitative dynamics of piecewise-linear differential equations: a discrete mapping approach. *Dyn. Syst.*, 4(3):565–583, 1989.

[11] E. H. Snoussi and R. Thomas. Logical identification of all steady states: the concept of feedback loop characteristic states. *Bull. Math. Biol.*, 55(5):973–991, 1993.

[12] R. Thomas and R. D'Ari. *Biological Feedback*. CRC Press.

[13] R. Thomas, D. Thieffry, and M. Kaufman. Dynamical behaviour of biological regulatory networks-I. Biological role of feedback loops and practical use of the concept of the loop characteristic state. *Bull. Math. Biol.*, 2(57):24–276, 1995.

[14] D. M. Wittmann, J. Krumsiek, J. Saez-Rodriguez, D. A. Lauffenburger, S. Klamt, and F. J. Theis. Transforming Boolean models to continuous models: methodology and application to T-cell receptor signaling. *BMC Systems Biology*, 3:98, 2009.

# Chapter 7

# Temporal Constraints in the Logical Analysis of Regulatory Networks

Heike Siebert and Alexander Bockmayr.

*Theoretical Computer Science*, 391(3):258-275, 2008.

**Remarks.** Alexander Bockmayr brought up the interesting question of how to incorporate information on time delays into a discrete model of a regulatory network. He proposed to make use of the theory of hybrid, and in particular timed, automata to enrich the discrete model with a continuous time evolution. I developed the modeling formalism as it is presented in this paper, proved the relation of the new formalism with the underlying Thomas formalism and modeled and analyzed the two example networks included in the paper. We worked jointly on the final version of the article.

# Temporal Constraints in the Logical Analysis of Regulatory Networks

Heike Siebert and Alexander Bockmayr

*Abstract.* Starting from the logical description of gene regulatory networks developed by R. Thomas, we introduce an enhanced modeling approach based on timed automata. We obtain a refined qualitative description of the dynamical behavior by exploiting not only information on ratios of kinetic parameters related to synthesis and decay, but also constraints on the time delays associated with the operations of the system. We develop a formal framework for handling such temporal constraints using timed automata, discuss the relationship with the original Thomas formalism, and demonstrate the potential of our approach by analyzing an illustrative gene regulatory network of bacteriophage $\lambda$.

## 1 Introduction

Gene regulatory networks can be modeled in various ways. Traditionally, such a system is modeled with differential equations. The equations used, however, are mostly non-linear and thus cannot be solved analytically. Furthermore, the available experimental data is often of qualitative character and does not allow a precise determination of quantitative parameters for the differential model. This eventually led to the development of qualitative modeling approaches in the work of Sugita, Kauffman, Glass, and Thomas [11, 6, 5, 13].. R. Thomas introduced a logical formalism in the 1970s, which, over the years, has been further developed and successfully applied to different biological problems (see [14], [15] and references therein). The only information on a concentration of gene products required in this formalism is whether or not it is above a threshold relevant for some interaction in the network. Furthermore, parameters holding information about the ratio of production and spontaneous decay rates of the gene products are used. The values of these parameters determine the dynamical behavior of the system, which is represented as a state transition graph. Moreover, Thomas realized that a realistic model should not be based on the assumption that the time delay from the start of the synthesis of a given product until the point where the concentration reaches a threshold is the same for all the genes in the network. Neither will the time delays associated with synthesis and those associated with decay be the same. Therefore, he uses an asynchronous description of the dynamics of the system, i. e., a state in the state transition graph differs from its predecessor in one component only.

In order to refine the model, we would like to incorporate information about the values of the time delays. Since precise data about the time delays is not available (in biological systems the delays will not even have an exactly determined value), the information is given in the form of inequalities that impose constraints on the time delays. So we need to keep

track of time while the system evolves. A theoretical framework providing us with the necessary concepts is the theory of timed automata. Each gene is equipped with a clock which is used to evaluate the conditions imposed on the time delays of that particular gene during the evolution of the system. The resulting transition system is in general nondeterministic, but the additional information inserted allows for a refined view of the dynamics. Conclusions about stability of dynamical behavior and restriction to certain behavior in comparison to the predictions of the Thomas model become possible. Also, the possibility of synchronous update is not excluded under certain conditions. Furthermore, our modeling approach permits the modeling of context sensitive systems. That is, interactions between the network's components are allowed to be of different character, i. e., inhibiting or activating, depending on the state of the system. The resulting framework is substantially less restrictive than the classical Thomas formalism.

The organization of this paper is as follows. We start in Sect. 2 with a mathematical presentation of the Thomas formalism, followed by a short review of the basic concepts of timed automata in Sect. 3. In Sect. 4 we develop our new modeling framework, which is the main contribution of this paper. In Sect. 5, we show that using our approach, it is possible to obtain the state transition graph of the original Thomas model. To illustrate the theoretical considerations, we analyze in Sect. 6 two regulatory networks of bacteriophage $\lambda$. The corresponding models have been implemented using the verification tool UPPAAL. In the last section, we discuss the mathematical and biological perspectives of our approach.

This is an extended version of a paper presented at CMSB'2006 [9].

## 2 Generalized Logical Formalism of Thomas

In this section we give a formal definition of a gene regulatory network in the sense of the modeling approach of R. Thomas (see for example [14] and [15]). We use mainly the formalism introduced in [4]. Throughout the text, $\{k, \ldots, l\}$, $k, l \in \mathbb{N}_0$, denotes the set $\{m \in \mathbb{N}_0 \mid k \leq m \leq l\}$.

### 2.1 Structure and Dynamics

To fully describe a gene regulatory network comprising $n$ genes $\alpha_1, \ldots, \alpha_n$ we have to take several steps. First, we describe the structure of the network by means of a labeled directed graph. In the following we use standard concepts from graph theory like e. g. paths and cycles.

**Definition 2.1.** *An* interaction graph *is a tuple* $\mathcal{I} = (G, \varepsilon, b, p)$ *where*

- $G = (V, E)$ *is a directed graph with vertex set* $V := \{\alpha_1, \ldots, \alpha_n\}$, $n \in \mathbb{N}$, *and edge set* $E \subset V \times V$,
- $\varepsilon : E \to \{+, -\}$,
- $b : E \to \mathbb{N}$, *and*
- $p : V \to \mathbb{N}, \alpha_i \mapsto \max\{b((\alpha_j, \alpha_i)) \mid j \in \{1, \ldots, n\}, \alpha_j \in Pred(\alpha_i)\}$, *where* $Pred(\alpha_i) := \{\alpha_j \in V \mid (\alpha_j, \alpha_i) \in E\}$ *is the set of* predecessors *of* $\alpha_i$ *for all* $i \in \{1, \ldots, n\}$. *We call the set* $\{k \in \mathbb{N}_0 \mid k \leq p(\alpha_i)\}$ *the* range *of* $\alpha_i$.

*For all* $i, j \in \{1, \ldots, n\}$, *an edge* $(\alpha_j, \alpha_i)$ *is also denoted by* $\alpha_j \to \alpha_i$, *and we set* $\varepsilon_{ij} = \varepsilon((\alpha_j, \alpha_i))$, $b_{ij} = b((\alpha_j, \alpha_i))$ *and* $p_i = p(\alpha_i)$.

The vertices of this graph represent the genes of the regulatory network, the range of a vertex the different expression levels of a gene affecting the behavior. An edge $\alpha_j \to \alpha_i$ signifies that the gene product of $\alpha_j$ influences the gene $\alpha_i$ in a positive or negative way depending on the sign $\varepsilon_{ij}$ and provided that the expression level of $\alpha_j$ is equal or above a threshold value which is given by $b_{ij}$. Note that the values $b_{ij}$ do not have to be pairwise distinct.

**Example 2.2.** *Fig. 1 shows a simplified model of a genetic network associated with the virus bacteriophage λ (see [12]). It comprises two genes $\alpha_1$, representing a bacteriophage gene called cI, and $\alpha_2$, representing a gene called cro, that influence each other as well as themselves. Thus the interaction graph of the model contains four edges, $(\alpha_1, \alpha_1)$, $(\alpha_1, \alpha_2)$, $(\alpha_2, \alpha_1)$ and $(\alpha_2, \alpha_2)$. The reason why the edge from $\alpha_1$ to itself is dotted will be explained later in Example 2.4. Only the loop containing $\alpha_1$ represents an activating influence and is thus labeled with a positive sign. All the other interactions of the network are of inhibiting character, which is represented by a negative sign. Experimental data shows that low concentrations of the product of $\alpha_2$ suffice for the inhibition of $\alpha_1$, while higher concentrations are necessary for the negative influence of $\alpha_2$ on itself to take effect. Therefore we choose threshold values $b_{12} = 1$ and $b_{22} = 2$. The range of $\alpha_2$ in this example is $\{0, 1, 2\}$, representing the set of expression levels of the gene $\alpha_2$ influencing the behavior of the system. Note that we only have to satisfy $b_{12} \leq b_{22}$ to be in agreement with our experimental observations. For example we could have chosen $b_{12} = 1$ and $b_{22} = 3$. However, this would result in a range $\{0, 1, 2, 3\}$ of $\alpha_2$, including expression levels superfluous for the modeling of the system's behavior. (If the gene expression level of $\alpha_2$ is 2, $\alpha_2$ has a negative influence on $\alpha_1$ but does not yet inhibit itself. The same is true if $\alpha_2$ has expression level 1.)*

*In contrast, $\alpha_1$ influences $\alpha_2$ as well as itself at the same expression level, so both edges starting in $\alpha_1$ are labeled by 1. The range of $\alpha_1$ is $\{0, 1\}$.*

In the next step we want to capture the dynamics of the gene regulatory network, i. e., the way the system behaves if we assign a specific expression level to each gene. We can interpret the genes $\alpha_i$ as variables that take values in the corresponding range, and an $n$-tuple of expression levels as a state of the system. The information captured in the interaction graph allows us to determine the active interactions in a given state by comparing the expression level of a gene with the thresholds labeling the edges originating in that gene. However, we have not yet specified how a given set of active interactions targeting the same gene influences the expression level of that gene.

**Definition 2.3.** *Let $\mathcal{I} = (G, \varepsilon, b, p)$ be an interaction graph of a gene regulatory network with $n$ genes $\alpha_1, \ldots, \alpha_n$. A* state *of the network described by $\mathcal{I}$ is a tuple $s = (s_1, \ldots, s_n) \in S := \{0, \ldots, p_1\} \times \cdots \times \{0, \ldots, p_n\}$. For all $i \in \{1, \ldots, n\}$, we denote by $s_i$ the $i$-th component of the state $s$. The set of* resources *$R_i(s)$ of $\alpha_i$ in state $s$ is the set*

$$R_i(s) := \{\alpha_j \in Pred(\alpha_i) \mid (\varepsilon_{ij} = + \wedge s_j \geq b_{ij}) \vee (\varepsilon_{ij} = - \wedge s_j < b_{ij})\}.$$

*Finally, we define the set of* (logical) parameters

$$K(\mathcal{I}) := \{K_{\alpha_i,\omega} \in \{0, \ldots, p_i\} \mid i \in \{1, \ldots, n\}, \omega \subseteq Pred(\alpha_i)\}.$$

*We call the pair $(\mathcal{I}, K(\mathcal{I}))$ a* gene regulatory network.

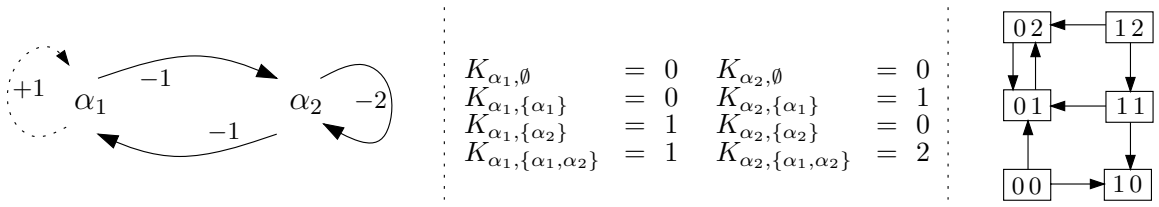



Figure 1: Interaction graph, parameters and state transition graph of a gene regulatory network associated with bacteriophage $\lambda$.

The set of resources $R_i(s)$ provides information about the presence of activators and the absence of inhibitors for some gene $\alpha_i$ in state $s$. The value of the parameter $K_{\alpha_i,R_i(s)}$ indicates how the expression level of gene $\alpha_i$ will evolve depending on $R_i(s)$. The product concentration will increase (resp. decrease) if the parameter value is greater (resp. smaller) than the current value $s_i$. The expression level stays the same if both values are equal.

**Example 2.4.** *In Fig. 1 we give a list of parameters specifying the behavior of the gene regulatory network described by the interaction graph from Example 2.2. Experimental observations are used to choose the different parameter values associated with each gene $\alpha_i$, $i \in \{1,2\}$, and the subsets of predecessors of $\alpha_i$ in the interaction graph (see [12] for details). To illustrate the meaning of the parameter values, let us consider the following example. Assume the system is in a state where gene $\alpha_1$ has no activating influence on itself, i. e., $\alpha_1$ has expression level 0, and is furthermore inhibited by $\alpha_2$. This situation can be represented by the state $(0,1)$ as well as $(0,2)$. We have $R_1((0,1)) = R_1((0,2)) = \emptyset$. Thus, regardless of the specific state, there is no influence on $\alpha_1$ that would indicate an activation of $\alpha_1$. The expression level should remain 0. This is reflected in the choice $K_{\alpha_1,\emptyset} = 0$. In the state $(0,1)$ we have $R_2((0,1)) = \{\alpha_1,\alpha_2\}$, since the expression level neither of $\alpha_1$ nor of $\alpha_2$ is high enough to effect inhibition. Experimental data suggests that in this case $\alpha_2$ will reach its highest expression level, thus we set $K_{\alpha_2,\{\alpha_1,\alpha_2\}} = 2$. Note that the state $(0,2)$ does not yield the same set of resources for $\alpha_2$, $R_2((0,2)) = \{\alpha_1\}$, and is associated with the logical parameter $K_{\alpha_2,\{\alpha_1\}} = 1$. That is, the states $(0,1)$ and $(0,2)$ represent the same situation with regard to the gene $\alpha_1$ but not with regard to $\alpha_2$.*

*A closer look of the parameter values associated with $\alpha_1$ shows that they render the loop starting in $\alpha_1$ ineffective with respect to the dynamics of the system. The parameter values do not depend on whether or not $\alpha_1$ has an activating influence on itself. This is why the loop $\alpha_1 \to \alpha_1$ has been drawn with a dotted line.*

The logical parameters allow us to derive the dynamics of the gene regulatory network, which we represent again by a directed graph. We define $sgn : \mathbb{Z} \to \{-1,0,1\}$ with $sgn(z) = -1$ for all $z < 0$, $sgn(0) = 0$, and $sgn(z) = 1$ for all $z > 0$.

**Definition 2.5.** *The* state transition graph $\mathcal{S}_N$ *corresponding to a gene regulatory network $N = (\mathcal{I}, K(\mathcal{I}))$ is a directed graph with vertex set $S$ as defined in Def. 2.3. There is an edge $s \to s'$ if there is $i \in \{1,\ldots,n\}$ such that the components of $s$ and $s'$ satisfy $s'_i = s_i + sgn(K_{\alpha_i,R_i(s)} - s_i) \neq s_i$ and $s_j = s'_j$ for all $j \in \{1,\ldots,n\} \setminus \{i\}$.*

The above definition reflects the use of the asynchronous update rule, since a state differs from a successor state in one component only. If $s$ is a state such that an evolution in more

than one component is indicated, then there will be more than one successor of $s$. Note that $s$ is a steady state if $s$ has no outgoing edge.

**Example 2.6.** *The state transition graph corresponding to the interaction graph and the parameter values discussed in the previous two examples is shown on the right of Fig. 1. Consider the state* $(0,0)$*. We first determine the set of resources of* $\alpha_1$ *and* $\alpha_2$ *in* $(0,0)$*, and obtain* $R_1((0,0)) = \{\alpha_2\}$ *and* $R_2((0,0)) = \{\alpha_1, \alpha_2\}$*. The parameter values* $K_{\alpha_1,\{\alpha_2\}} = 1$ *and* $K_{\alpha_2,\{\alpha_1,\alpha_2\}} = 2$ *given in Fig. 1 indicate an expression level change for both genes. However, in order to obtain a realistic representation of the dynamics, we take two considerations into account. First, changes of expression level happen gradually. Thus, although the parameter indicates an expression level increase from 0 to 2 for* $\alpha_2$*, we only allow a value change by 1 per step. Furthermore, we assume that the different biological processes underlying the expression level change from 0 to 1 for* $\alpha_1$ *and that for* $\alpha_2$ *will not take the exact same amount of time. One change will happen before the other. But since we do not know which process is faster, we include both possibilities in the state transition graph. We obtain two edges starting in the state* $(0,0)$*, representing the expression level change of* $\alpha_1$ *while fixing the expression level of* $\alpha_2$ *and vice versa. More precisely, we have the edges* $s = (0,0) \to (1,0) = s'$*, since* $s'_1 = 1 = 0 + sgn(1-0)$ *and* $s'_2 = s_2 = 0$*, and* $s = (0,0) \to (0,1) = s''$*, since* $s''_2 = 1 + sgn(2-0)$ *and* $s''_1 = s_1 = 0$*.*

*When analyzing the state transition graph we are mainly interested in asymptotical behavior. In our example in Fig. 1 we can see that the system will eventually reach and then remain in either the state* $(1,0)$ *or the cycle comprising the states* $(0,1)$ *and* $(0,2)$*. A biological interpretation of this result will be given in Sect. 6.1.*

## 2.2 Parameter Constraints

Thomas and Snoussi used their formalism to discretize a certain class of differential equation systems (see e.g. [10]). To reflect this, the following constraint has to be imposed on the parameter values:

$$\omega \subseteq \omega' \Rightarrow K_{\alpha_i,\omega} \leq K_{\alpha_i,\omega'} \tag{1}$$

for all $i \in \{1, \ldots, n\}$. This condition signifies that an effective activator or a non-effective inhibitor cannot induce the decrease of the expression level of $\alpha_i$. In biology, there are situations where this condition is not met. For instance, two substances can have activating properties when isolated from each other, but act inhibiting when combined. Thus it would be desirable to relax this restrictive condition.

The definition of the state transition graph representing the dynamics of a system does not have to be altered when dropping the parameter constraints (1). However, the interpretation of the corresponding interaction graph, and as a consequence that of the definition of the resources, becomes more difficult. Condition (1) ensures that the way two components influence each other, i. e., activation or inhibition, does not depend on the state of the system. For instance, if there is a positive edge from $\alpha_j$ to $\alpha_i$, then the increase of the expression level of $\alpha_j$ can never lead to a decrease of the expression level of $\alpha_i$. In this sense, condition (1) formalizes the intuitive interpretation of the signs in the interaction graph. A source of a positive (resp. negative) interaction will never act as an inhibitor (resp. activator) of the corresponding target. The interaction graph is global, not depending on the current state of the system.

When dropping the constraints (1) it is possible to model systems including components acting as both activators or inhibitors, depending on the state of the other components (as

mentioned above) or even depending on their own expression level (e. g. activating at low concentration levels, inhibiting at high levels). This is an important property for realistic modeling of many biological systems. In doing so, the signs in the interaction graph loose the interpretation mentioned above. They become a strictly technical tool for defining the resources and thus the logical parameters. A way to preserve the meaning of the signs in the interaction graph while dropping the parameter constraints (1) is to introduce local interaction graphs that describe the interactions occurring between the components of the system in a given state. By combining the local interaction graphs, a multigraph representing the network structure and character of the interactions can be derived. A notion of local interaction graphs has been proposed in [8].

The modeling approach we introduce in Sect. 4 is also of local character. Basically, we model each gene $\alpha_i$ on its own, incorporating conditions derived from network structure and parameter values that determine the behavior of $\alpha_i$ for every state of the system. In general, we do not impose the constraints (1). However, we will assume that (1) holds when comparing the dynamical behavior resulting from the original Thomas formalism and our approach.

## 3 Timed Automata

In this section we formally introduce timed automata. We mainly use the definitions and notations given in [1]. To introduce the concept of time in our system, we consider a set $C := \{c_1, \ldots, c_n\}$ of real variables that behave according to the differential equations $\dot{c_i} = 1$. These variables are called *clocks*. They progress synchronously and can be reset to zero under certain conditions. We define the set $\Phi(C)$ of *clock constraints* $\varphi$ by the grammar

$$\varphi ::= c \leq q \,|\, c \geq q \,|\, c < q \,|\, c > q \,|\, \varphi_1 \wedge \varphi_2 \,,$$

where $c \in C$ and $q$ is a rational constant.

A *clock interpretation* is a function $u : C \to \mathbb{R}_{\geq 0}$ from the set of clocks to the non-negative reals. For a given clock interpretation $u$ and $\delta \in \mathbb{R}_{\geq 0}$, we define the clock interpretation

$$u + \delta : C \to \mathbb{R}_{\geq 0}, \;\; c \mapsto u(c) + \delta \,.$$

Furthermore, for $R \subseteq C$, we define $u[R := 0] : C \to \mathbb{R}_{\geq 0}$ with $u[R := 0](c) = 0$ for all $c \in R$ and $u[R := 0](c) = u(c)$ for all $c \in C \setminus R$. We say that a clock interpretation $u$ *satisfies* a clock constraint $\varphi$ if $\varphi(u) = true$. The set of all clock interpretations is denoted by $\mathbb{R}^C_{\geq 0}$.

**Definition 3.1.** *A* timed automaton *$A$ is a tuple $A = (L, L^0, \Sigma, C, I, E)$, where*

- *$L$ is a finite set of objects called* locations*,*
- *$L^0 \subseteq L$ is a set the elements of which are called* initial locations*,*
- *$\Sigma$ is a finite set of objects called* events *(or labels),*
- *$C$ is a finite set of* clocks*,*
- *$I : L \to \Phi(C)$ is a mapping that labels each location with some clock constraint called the* invariant *of the location,*
- *and $E \subseteq L \times \Sigma \times \Phi(C) \times 2^C \times L$ is a set the elements of which are called* switches*. For each switch $e = (l, \sigma, \varphi, R, l')$ we call the clock constraint $\varphi$ the* guard *of $e$.*

A timed automaton can be represented as a directed graph with vertex set $L$. The vertices are labeled with the corresponding invariants and are marked as initial locations if they belong to $L^0$. The edges of the graph correspond to the switches and are labeled with an event, the guard specifying when the switch is enabled, and a subset of $C$ comprising the clocks that are reset to zero when the switch is executed. While switches are instantaneous, time may elapse in a location. To describe the dynamics of such an automaton formally, we use the notion of a transition system.

**Definition 3.2.** *Let $A = (L, L^0, \Sigma, C, I, E)$ be a timed automaton. The* (labeled) transition system *$T_A$ associated with $A$ is a tuple $(Q, Q^0, \Gamma, \to)$, where*

- $Q := \{(l, u) \in L \times \mathbb{R}^C_{\geq 0} \mid u \text{ satisfies } I(l)\}$,
- $Q^0 := \{(l, u) \in Q \mid l \in L^0, u(c) = 0 \text{ for all } c \in C\}$,
- $\Gamma := \Sigma \cup \mathbb{R}_{\geq 0}$,
- *and $\to \subseteq Q \times \Gamma \times Q$ is defined as the set comprising*
  - *$((l, u), \delta, (l, u + \delta))$ for all $(l, u) \in Q$ and $\delta \in \mathbb{R}_{\geq 0}$ such that for all $0 \leq \delta' \leq \delta$ the clock interpretation $u + \delta'$ satisfies the invariant $I(l)$,*
  - *and $((l, u), a, (l', u[R := 0]))$ for all $(l, u) \in Q$ and $a \in \Sigma$ such that there is a switch $(l, a, \varphi, R, l')$ in $E$, $u$ satisfies $\varphi$, and $u[R := 0]$ satisfies $I(l')$.*

*Often, we denote the tuple $((l, u), \gamma, (l', v)) \in \to$ by $(l, u) \xrightarrow{\gamma} (l', v)$. The elements of $Q$ are called* states *of $A$, the elements of $\to$ are called* transitions.

The first kind of transition is a state change due to elapse of time, while the second one is due to a location-switch and is called *discrete*. Again we can visualize the object $T_A$ as a directed graph with vertex set $Q$ and edges corresponding to the transitions given by $\to$. Note that by definition the set of states may be infinite and that the transition system is in general nondeterministic, i. e., a state may have more than one successor. Moreover, it is possible that a state is the source for edges labeled with a real value as well as for edges labeled with events. However, although every discrete transition corresponds to a switch in $A$, there may be switches in $A$ that do not lead to a transition in $T_A$. That is due to the additional conditions placed on the clock interpretations.

Finally, we obtain a modified transition system by considering only the location vectors as states, dropping all transitions labeled with real values, but keeping every discrete transition of $T_A$. We call this the *discrete (or symbolic) transition system* of $A$. Examples of timed automata and transition systems will be given in the next sections.

# 4 Modelling with Timed Automata

In order to model a gene regulatory network as a timed automaton, we first introduce components that correspond to the genes of the network. They constitute the building blocks that compose the automaton representing the network dynamics much in the same way $n$ timed automata are integrated into a product automaton (see [1]).

In the following, let $N = (\mathcal{I}, K(\mathcal{I}))$ be a gene regulatory network comprising the genes $\alpha_1, \ldots, \alpha_n$. Recall the terms $\varepsilon_{ij}$, $b_{ij}$, $p_i$ etc. introduced in the definitions given in Sect. 2. We will illustrate each step of the modeling process with the example introduced in Fig. 1.

## 4.1 Constructing the Components

We first give the formal definition for the component automata, adding illustrative remarks afterwards.

**Definition 4.1.** *A component automaton representing the gene $\alpha_i$, $i \in \{1, \ldots, n\}$, of the network $N$ is a timed automaton $A_i := (L_i, L_i^0, \Sigma_i, C_i, I_i, E_i)$ satisfying the following conditions.*

- *The set of locations $L_i$ contains objects $\alpha_i^k$ for $k \in \{0, \ldots, p_i\}$, $\alpha_i^{k+}$ for $k \in \{0, \ldots, p_i - 1\}$, and $\alpha_i^{k-}$ for $k \in \{1, \ldots, p_i\}$. The locations $\alpha_i^k$, $k \in \{0, \ldots, p_i\}$, are called* regular*, all other locations are called* intermediate*.*
- *The set $L_i^0$ of initial locations is defined as $L_i^0 := \{\alpha_i^k \mid k \in \{0, \ldots, p_i\}\}$.*
- *The set of events $\Sigma$ is given as $\Sigma_i := \{a_i^{k+}, a_i^{m-} \mid k \in \{0, \ldots, p_i - 1\}, m \in \{1, \ldots, p_i\}\}$.*
- *The set $C_i$ of clocks comprises a single clock $c_i$, i. e., $C_i := \{c_i\}$.*
- *Let $T_i^{k+}, T_i^{l-} \in \mathbb{Q}_{\geq 0}$ for all $k \in \{0, \ldots, p_i - 1\}$ and $l \in \{1, \ldots, p_i\}$.*
  *The mapping $I_i : L_i \to \Phi(C_i)$ satisfies $I_i(\alpha_i^k) = (c_i \geq 0)$ for all $k \in \{0, \ldots, p_i\}$, $I_i(\alpha_i^{k+}) = (c_i \leq T_i^{k+})$ for all $k \in \{0, \ldots, p_i - 1\}$, and $I_i(\alpha_i^{k-}) = (c_i \leq T_i^{k-})$ for all $k \in \{1, \ldots, p_i\}$.*
- *Let $t_i^{k+}, t_i^{l-} \in \mathbb{Q}_{\geq 0}$ for all $k \in \{0, \ldots, p_i - 1\}$, $l \in \{1, \ldots, p_i\}$. The set $E_i$ consists of the switches*

$$(\alpha_i^{k+}, a_i^{k+}, \varphi_i^{k+}, \{c_i\}, \alpha_i^{k+1}) \in L_i \times \Sigma_i \times \Phi(C_i) \times 2^{C_i} \times L_i\,,$$

  *where $\varphi_i^{k+} = (c_i \geq t_i^{k+})$ and $k \in \{0, \ldots, p_i - 1\}$, and the switches*

$$(\alpha_i^{l-}, a_i^{l-}, \varphi_i^{l-}, \{c_i\}, \alpha_i^{l-1}) \in L_i \times \Sigma_i \times \Phi(C_i) \times 2^{C_i} \times L_i\,,$$

  *where $\varphi_i^{l-} = (c_i \geq t_i^{l-})$ and $l \in \{1, \ldots, p_i\}$.*

In the following we motivate the above definition, considering one item at a time.

*Locations:* In our definition we distinguish between regular and intermediate locations. The regular location $\alpha_i^k$, $k \in \{0, \ldots, p_i\}$, represents a situation where gene $\alpha_i$ maintains expression level $k$, i. e., no expression level change is indicated. The location represents a momentary stable situation for the gene $\alpha_i$. The regular locations are also the initial locations.

In order to measure time delays, we need to know when a gene starts the process of increasing or decreasing its expression level. We achieve this by introducing the intermediate locations. In an intermediate location $\alpha_i^{k\varepsilon}$, $\varepsilon \in \{+, -\}$, the gene is in the process of changing its expression level. If the superscript is $k+$ (resp. $k-$), the expression level is $k$ but the concentration of the gene product increases (resp. decreases) with time. For example, the location $\alpha_1^{0+}$ represents the situation that gene $\alpha_1$ is in the process of changing its expression level from 0 to 1. Note that the expression level in this location is still 0, indicated by the number in the superscript.

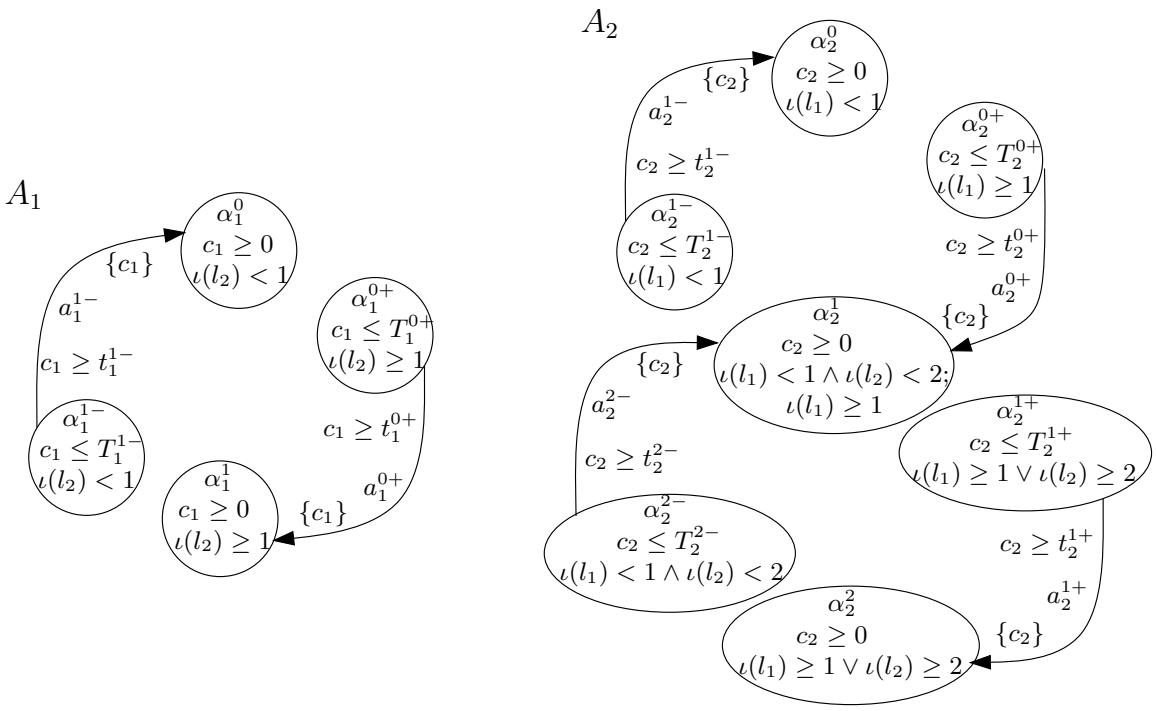



Figure 2: Components $A_1$ and $A_2$ representing the genes $\alpha_1$ and $\alpha_2$ in Fig. 1.

**Example 4.2.** *As shown in Fig. 2, the component $A_1$ corresponding to gene $\alpha_1$ of the example in Fig. 1 contains two regular locations signifying its expression levels 0 and 1. Component $A_2$ has three regular locations, namely $\alpha_2^0$, $\alpha_2^1$ and $\alpha_2^2$, since three distinct expression levels are associated with gene $\alpha_2$. Accordingly, $A_2$ has four intermediate locations while $A_1$ has only two.*

*Events:* The events in $\Sigma_i$ correspond to the intermediate locations. These events will be used later on to identify certain discrete transitions starting in the intermediate locations, namely those that result in a change of expression level. For example, the event $a_1^{0+}$ signifies that the expression level of gene $\alpha_1$ changes from zero to one.

*Clocks:* Each gene $\alpha_i$ is equipped with a single clock $c_i$ to measure the duration of the expression level changes.

*Invariants:* Every regular location $\alpha_i^k$ is mapped by $I_i$ to $(c_i \geq 0)$ (evaluating to *true*). That is, the question whether or not the system remains in a regular location does not depend on the clock values. Now, we make the first step in incorporating time delays. Since it is not realistic to assign an exact time delay to a biological process such as change of expression level, we rather use an interval bounded by a maximal and minimal time delay. Every intermediate location $\alpha_i^{k\varepsilon}$, $\varepsilon \in \{+, -\}$, is associated with an invariant $(c_i \leq T_i^{k\varepsilon})$. The value $T_i^{k\varepsilon}$ signifies the maximal time delay before the expression level of $\alpha_i$ changes to $k+1$, if $\varepsilon = +$, or to $k-1$, if $\varepsilon = -$.

*Switches:* We use the guards of the switches to include the minimal time delay of a process of expression level change. There are two kinds of switches in the set $E_i$. For all $k \in \{0, \ldots, p_i - 1\}$, we have $(\alpha_i^{k+}, a_i^{k+}, \varphi_i^{k+}, \{c_i\}, \alpha_i^{k+1}) \in E_i$, where $\varphi_i^{k+} = (c_i \geq t_i^{k+)})$, representing increase of expression level. Furthermore, for $l \in \{1, \ldots, p_i\}$, the switch $(\alpha_i^{l-}, a_i^{l-}, \varphi_i^{l-}, \{c_i\}, \alpha_i^{l-1})$ with

$\varphi_i^{l-} = (c_i \geq t_i^{l-})$ belongs to $E_i$ and represents expression level decrease. The given time constraints determine the minimal time delay before a change in expression level can occur. Choosing the time constants associated with the guards strictly smaller than those associated with the invariants of the corresponding intermediate location leads to indeterministic behavior of the system in this location. We reset the automaton clock $c_i$ to 0 after every location change as indicated by the set $\{c_i\}$ in the fourth component of the switch tuple.

**Example 4.3.** *Fig. 2 shows the graph representation of the timed automata $A_1$ and $A_2$ corresponding to the genes $\alpha_1$ and $\alpha_2$ introduced in Fig. 1. The locations depicted as ellipses are labeled with the location name, the corresponding invariant, and further information that will be explained below. The switches are represented as edges between locations labeled with the corresponding guard, event and the set containing the clock of the automaton.*

In a last step, we have to incorporate information of network interactions and parameters. We label each location with conditions concerning the expression levels of the interacting genes. If the conditions are met, a change in the location is indicated. We call these conditions *switch conditions*. Note that in general the conditions can only be evaluated in the network context, since information about the current location of all interacting genes is needed. Again, we first give the formal definition which we motivate afterwards with a more intuitive description. Recall the notations $\varepsilon_{ij}$, $b_{ij}$, $K_{\alpha_i,\omega}$ etc. from Sect. 2.

**Definition 4.4.** *Let $A_i := (L_i, L_i^0, \Sigma_i, C_i, I_i, E_i)$, $i \in \{1,\dots,n\}$, be the component automata representing the genes of the network $N$. Let $\iota : \bigcup_{j\in\{1,\dots,n\}} L_j \to \mathbb{N}_0$ such that $\iota(\alpha_j^k) = \iota(\alpha_j^{k+}) = k$ for all $j \in \{1,\dots,n\}$ and $k \in \{0,\dots,p_j-1\}$, and $\iota(\alpha_j^k) = \iota(\alpha_j^{k-}) = k$ for all $j \in \{1,\dots,n\}$ and $k \in \{1,\dots,p_j\}$.*

1. *For every $i,j \in \{1,\dots,n\}$ such that $\alpha_j \in Pred(\alpha_i)$ and $l_j \in L_j$, set*

$$\lambda_i^{\alpha_j}(l_j) := \begin{cases} \iota(l_j) \geq b_{ij}\,, & \varepsilon_{ij} = + \\ \iota(l_j) < b_{ij}\,, & \varepsilon_{ij} = - \end{cases}, \quad \overline{\lambda}_i^{\alpha_j}(l_j) := \begin{cases} \iota(l_j) < b_{ij}\,, & \varepsilon_{ij} = + \\ \iota(l_j) \geq b_{ij}\,, & \varepsilon_{ij} = - \end{cases}.$$

2. *For every $i \in \{1,\dots,n\}$ and $k \in \{0,\dots,p_i-1\}$, let*

$$\omega_1^{i,k},\dots,\omega_{m_1^{i,k}}^{i,k} \subseteq Pred(\alpha_i)\,,$$

*$m_1^{i,k} \in \mathbb{N}$, be the subsets of $Pred(\alpha_i)$ such that $K_{\alpha_i,\omega_h} > k$ for $h \in \{1,\dots,m_1^{i,k}\}$. For every $i \in \{1,\dots,n\}$ and $k \in \{1,\dots,p_i\}$, let*

$$\upsilon_1^{i,k},\dots,\upsilon_{m_2^{i,k}}^{i,k} \subseteq Pred(\alpha_i)\,,$$

*$m_s^{i,k} \in \mathbb{N}$, be the subsets of $Pred(\alpha_i)$ such that $K_{\alpha_i,\upsilon_h} < k$ for $h \in \{1,\dots,m_2^{i,k}\}$.*

3. *For $l \in L_1 \times \cdots \times L_n$, $i \in \{1,\dots,n\}$, $k \in \{0,\dots,p_i\}$, and $\rho \in \{\omega_1^{i,k},\dots,\omega_{m_1^{i,k}}^{i,k}, \upsilon_1^{i,k},\dots,\upsilon_{m_2^{i,k}}^{i,k}\}$, set*

$$\lambda_i^\rho(l) := \bigwedge_{\alpha_j\in\rho} \lambda_i^{\alpha_j}(l_j) \quad \wedge \bigwedge_{\alpha_j\in Pred(\alpha_i)\setminus\rho} \overline{\lambda}_i^{\alpha_j}(l_j).$$

*4. For every* $l \in L_1 \times \cdots \times L_n$ *and* $i \in \{1, \ldots, n\}$, *set*

$$\Lambda_i^{k+}(l) := \bigvee_{h \in \{1,\ldots,m_1^{i,k}\}} \lambda_i^{\omega_h^{i,k}}(l) \qquad \text{for } k \in \{0, \ldots, p_i - 1\},$$

*and*

$$\Lambda_i^{k-}(l) := \bigvee_{h \in \{1,\ldots,m_2^{i,k}\}} \lambda_i^{\upsilon_h^{i,k}}(l) \qquad \text{for } k \in \{1, \ldots, p_i\}\,.$$

*We call* $\Lambda_i^{k+}(l)$ *and* $\Lambda_i^{k-}(l)$ switch conditions for $A_i$ in $l$.

In the Thomas formalism introduced in Sect. 2 we derive the system's dynamical behavior in a given state by considering for each gene the current expression level, the set of resources in the state and the corresponding parameter value. The same considerations lead to the definition of the switch conditions for the component automata. So let us fix a tuple $l = (l_1, \ldots, l_n) \in L_1 \times \cdots \times L_n$. The first, most basic step is to derive the expression level of a gene $\alpha_i$ from the location $l_i$ of the component automaton $A_i$. This is achieved by applying the mapping $\iota$ to the location, in accordance with the interpretation of the locations given above. Thus we can associate a state $s := (\iota(l_1), \ldots, \iota(l_n))$ of the system $N$ with $l$. In a next step, we determine for each $\alpha_i$ the resources in the state $s$ by verifying the expressions given in *1.* in the above definition. If $\lambda_i^{\alpha_j}(l_j)$, $\alpha_j \in Pred(\alpha_i)$, evaluates to *true*, then $A_j$ represents a resource of $\alpha_i$ in the given location. If the negation is true, then $A_j$ does not represent a resource.

**Example 4.5.** *Consider again our running example from Fig. 1 and 2, and the location pair* $(\alpha_1^{0+}, \alpha_2^1)$. *The pair represents the situation of gene* $\alpha_1$ *having expression level* $\iota(\alpha_1^{0+}) = 0$, *and gene* $\alpha_2$ *having expression level* $\iota(\alpha_2^1) = 1$. *In addition, we know from the location identifiers that* $\alpha_1$ *is in the process of increasing its expression level while* $\alpha_2$ *is currently stable. This additional information however is of no consequence at the moment. Now, we construct the expressions given in 1. in Def. 4.4 for an arbitrary location pair* $(l_1, l_2) \in L_1 \times L_2$. *For* $A_1$ *we obtain* $\lambda_1^{\alpha_1}(l_1) = (\iota(l_1) \geq 1)$, $\lambda_1^{\alpha_2}(l_2) = (\iota(l_2) < 1)$ *and the two negations* $\overline{\lambda}_1^{\alpha_1}(l_1) = (\iota(l_1) < 1)$ *and* $\overline{\lambda}_1^{\alpha_2}(l_2) = (\iota(l_2) \geq 1)$. *For* $A_2$ *we derive* $\lambda_2^{\alpha_1}(l_1) = (\iota(l_1) < 1)$, $\lambda_2^{\alpha_2}(l_2) = (\iota(l_2) < 2)$ *and the negations* $\overline{\lambda}_2^{\alpha_1}(l_1)$ *and* $\overline{\lambda}_2^{\alpha_2}(l_2)$. *For our chosen location pair we see that, for example,* $\lambda_2^{\alpha_1}(\alpha_1^{0+}) = (0 < 1) = \mathit{true}$ *and* $\lambda_2^{\alpha_2}(\alpha_2^1) = (1 < 2) = \mathit{true}$. *Thus, both* $\alpha_1$ *and* $\alpha_2$ *are resources of* $\alpha_2$ *in the state of the system represented by* $(\alpha_1^{0+}, \alpha_2^1)$.

In order to find out whether or not a location change in $A_i$ is indicated in the state represented by $l$, we have to consider the parameter values that determine the dynamics of the system. Therefore, we determine all sets $\rho \subseteq Pred(\alpha_i)$ such that either $K_{\alpha_i,\rho} > k$ or $K_{\alpha_i,\rho} < k$, where $k$ is the current expression level of $\alpha_i$, i. e., $\iota(l_i) = k$. This is formalized in *2.* in Def. 4.4.

**Example 4.6.** *In our example from Fig. 1 we obtain for* $\alpha_2$ *and* $k = 1$ *the sets* $\omega_1 := \omega_1^{2,1} = \{\alpha_1, \alpha_2\}$, $\upsilon_1 := \upsilon_1^{2,1} = \emptyset$ *and* $\upsilon_2 := \upsilon_2^{2,1} = \{\alpha_2\}$.

Now, we just have to check if the set of resources we determined for $\alpha_i$ in the given state $s$ matches one of the sets $\rho$. This is achieved by evaluating the expression $\lambda_i^{\rho}(l)$ given in *3.* in Def. 4.4 for each $\rho$. If $\lambda_i^{\omega_h^{i,k}}(l)$ is true for some $\omega_h^{i,k}$ as defined above, then an increase of expression level of gene $\alpha_i$ is indicated. If $\lambda_i^{\upsilon_h^{i,k}}(l)$ is satisfied for some $\upsilon_h^{i,k}$, then the expression level will decrease. Note that the definition ensures that we have no conflicting commands, i. e., increase and decrease, for $\alpha_i$ in $l$.

**Example 4.7.** *For our running example and with $\omega_1$, $\upsilon_1$ and $\upsilon_2$ as calculated above we obtain $\lambda_2^{\omega_1}(l) = (\iota(l_1) < 1) \wedge (\iota(l_2) < 2)$, $\lambda_2^{\upsilon_1}(l) = (\iota(l_1) \geq 1) \wedge (\iota(l_2) \geq 2)$ and $\lambda_2^{\upsilon_2}(l) = (\iota(l_1) \geq 1) \wedge (\iota(l_2) < 2)$. Thus, if the system is in state $l' = (\alpha_1^{0+}, \alpha_2^1)$, for instance, the condition $\lambda_2^{\omega_1}(l')$ is true and the expression level of $\alpha_2$ should increase, while for $l = (\alpha_1^1, \alpha_2^1)$ condition $\lambda_2^{\upsilon_2}(l)$ is satisfied and indicates expression level decrease. Note that condition $\lambda_2^{\upsilon_1}(l)$ is not satisfied.*

In order to induce a corresponding change in expression level, it is sufficient if the condition $\lambda_i^{\omega_h^{i,k}}(l)$ resp. $\lambda_i^{\upsilon_h^{i,k}}(l)$ holds for some $\omega_h^{i,k}$ resp. $\upsilon_h^{i,k}$. Due to this observation we set the switch conditions as disjunctions, as formulated in *4.* in Def 4.4.

Now, we assign all locations $\alpha_i^k$, $k \in \{1, \ldots, p_i - 1\}$ the conditions $\Lambda_i^{k+}(l)$ and $\Lambda_i^{k-}(l)$ with $l \in L_1 \times \cdots \times L_n$. The location $\alpha_i^0$ resp. $\alpha_i^{p_i}$ is labeled with $\Lambda_i^{0+}(l)$ resp. $\Lambda_i^{p_i-}(l)$ only, since the location represents the lowest resp. highest expression level possible. Furthermore, we want to check in an intermediate location whether the condition that led to the process of changing the expression level is still valid. If that is not the case, the system should not remain in that location Thus, we associate with location $\alpha_i^{k+}$ the condition $\neg\Lambda_i^{k+}(l)$ for all $k \in \{0, \ldots, p_i - 1\}$, and allot to location $\alpha_i^{k-}$ the condition $\neg\Lambda_i^{k-}(l)$ for all $k \in \{1, \ldots, p_i\}$. Thus for every given location tuple $l$ we have to check the switch conditions associated with $l_i$ to determine the course of behavior for the component automaton $A_i$.

All the above considerations on how the switch conditions should influence the behavior of the system are of no consequence for the definition of the component automata. Here, we only need to formulate the switch conditions according to the data given by the gene regulatory network $N$ and assign them to the appropriate locations. The desired impact of the switch conditions on the behavior of the system will be realized in the definition of the timed automaton representing the network dynamics.

Formally speaking, the components defined above are timed automata. However, it does not make sense to evaluate their behavior in isolation from each other. This becomes apparent when looking at the graph representation. Most locations in the automaton $A_i$ are not connected by edges. Every path in the graph contains at most one edge. Fig. 2 illustrates this observation. The behavior of the gene regulatory network is captured when allowing the components to interact, the rules of interaction being derived from the switch conditions.

## 4.2 Simplifying the Switch Conditions

The definition of the switch conditions reflects the local character of our modeling approach. The boolean value resulting from the evaluation of the switch conditions obviously depends on the state the system is in. This allows for a flexible description of interactions between the components of the network as already discussed in Sect. 2.2.

However, given a concrete model, it is often possible to simplify the switch conditions. Assume, for instance, that the condition (1) introduced in Sect. 2.2 holds and define sets $\omega_1^{i,k}, \ldots, \omega_{m_1^{i,k}}^{i,k}, \upsilon_1^{i,k}, \ldots, \upsilon_{m_2^{i,k}}^{i,k}$ as in the preceding section. To simplify notation we drop the superscript $i, k$ in this subsection. Then it is sufficient to define $\lambda_i^{\omega_h}(l) := \bigwedge_{\alpha_j \in \omega_h} \lambda_i^{\alpha_j}(l_j)$, since the addition of another resource never results in a smaller parameter value. An increase of expression level is indicated if $\bigwedge_{\alpha_j \in \omega_h} \lambda_i^{\alpha_j}(l_j)$ is true regardless of the expression level of predecessors of $\alpha_i$ not contained in $\omega_h$. For the same reason we can define $\lambda_i^{\upsilon_h}(l) := \bigwedge_{\alpha_j \in Pred(\alpha_i) \setminus \upsilon_h} \overline{\lambda}_i^{\alpha_j}(l_j)$.

Moreover, whenever $\omega_{h_1} \subseteq \omega_{h_2}$ for sets $\omega_h$, then $\lambda_i^{\omega_{h_1}}(l)$ is true if $\lambda_i^{\omega_{h_2}}(l)$ is true. Since

condition (1) implies that $K_{\alpha_i,\omega_{h_2}} \geq K_{\alpha_i,\omega_{h_1}} > k$, we can delete condition $\lambda_i^{\omega_{h_2}}(l)$ from the expression $\Lambda_i^{k+}(l)$. Analogously, if $\upsilon_{h_1} \subseteq \upsilon_{h_2}$, we can delete the condition $\lambda_i^{\upsilon_{h_1}}(l)$ from the expression $\Lambda_i^{k-}(l)$. Fig. 2 shows the components $A_1$ and $A_2$ corresponding to the genes of our running example, which satisfies condition (1). The switch conditions are given in the simplified form explained above.

In general, any inequality concerning the expression level of the location the inequality is associated with can be evaluated immediately. For example, the location $\alpha_2^2$ in Fig. 2 is labeled with the switch condition $\iota(l_1) \geq 1 \vee \iota(l_2) \geq 2$. Since $\iota(\alpha_2^2) \geq 2$, the switch condition in that location is always true, regardless of the state of $A_1$.

### 4.3 Modelling the Network

Combining the components $A_1, \ldots, A_n$, we now construct the timed automaton $A_N$ representing the whole network $N$.

**Definition 4.8.** *Let $A_i := (L_i, L_i^0, \Sigma_i, C_i, I_i, E_i)$, $i \in \{1, \ldots, n\}$, be the component automata representing the genes of the network $N$. The* network automaton *representing $N$ is the timed automaton $A_N := (L, L^0, \Sigma, C, I, E)$ where*

- $L := L_1 \times \cdots \times L_n$,
- $L^0 := L_1^0 \times \cdots \times L_n^0$,
- $\Sigma := \{a\} \cup \bigcup_{i\in\{1,\ldots,n\}} \Sigma_i$ *with $a$ being an object satisfying $a \notin \bigcup_{i\in\{1,\ldots,n\}} \Sigma_i$,*
- $C := \bigcup_{i\in\{1,\ldots,n\}} C_i$,
- $I : L \to \Phi(C)$, $(l_1, \ldots, l_n) \mapsto (I_1(l_1) \wedge \cdots \wedge I_n(l_n))$,
- *and the set of switches $E \subseteq L \times \Sigma \times \Phi(C) \times 2^C \times L$ is comprised of the following elements:*
  - *For $i \in \{1, \ldots, n\}$ and every switch $(l_i, a_i, \varphi_i, R_i, l_i') \in E_i$ the tuple $(h, a_i, \varphi_i, R_i, h')$, with $h, h' \in L$, $h_j = h_j'$ for all $j \neq i$, $h_i = l_i$ and $h_i' = l_i'$, is a switch in $E$. That is, we preserve the switches of the components.*
  - *Let $l \in L$. Let $J$ be the set of those $j \in \{1, \ldots, n\}$ such that for each $l_j, j \in J$ one of the associated switch conditions is true. Assume $R$ comprises the clocks $c_j$, $j \in J$. Let $l_i = l_i'$ for all $i \notin J$, and let, for all $j \in J$,*

$$l_j' = \begin{cases} \alpha_j^{k-}, \text{ if } l_j = \alpha_j^k & \text{for some } k \text{ and } \Lambda_j^{k-}(l) = \text{true} \\ \alpha_j^{k+}, \text{ if } l_j = \alpha_j^k & \text{for some } k \text{ and } \Lambda_j^{k+}(l) = \text{true} \\ \alpha_j^k \text{ , if } l_j = \alpha_j^{k-} & \text{for some } k \text{ and } \neg\Lambda_j^{k-}(l) = \text{true} \\ \alpha_j^k \text{ , if } l_j = \alpha_j^{k+} & \text{for some } k \text{ and } \neg\Lambda_j^{k+}(l) = \text{true} \end{cases} \tag{2}$$

  *Then $(l, a, \text{true}, R, l')$ is a switch in $E$.*

*A location in $L$ is called* regular*, if all of its components are regular, and* intermediate *otherwise.*

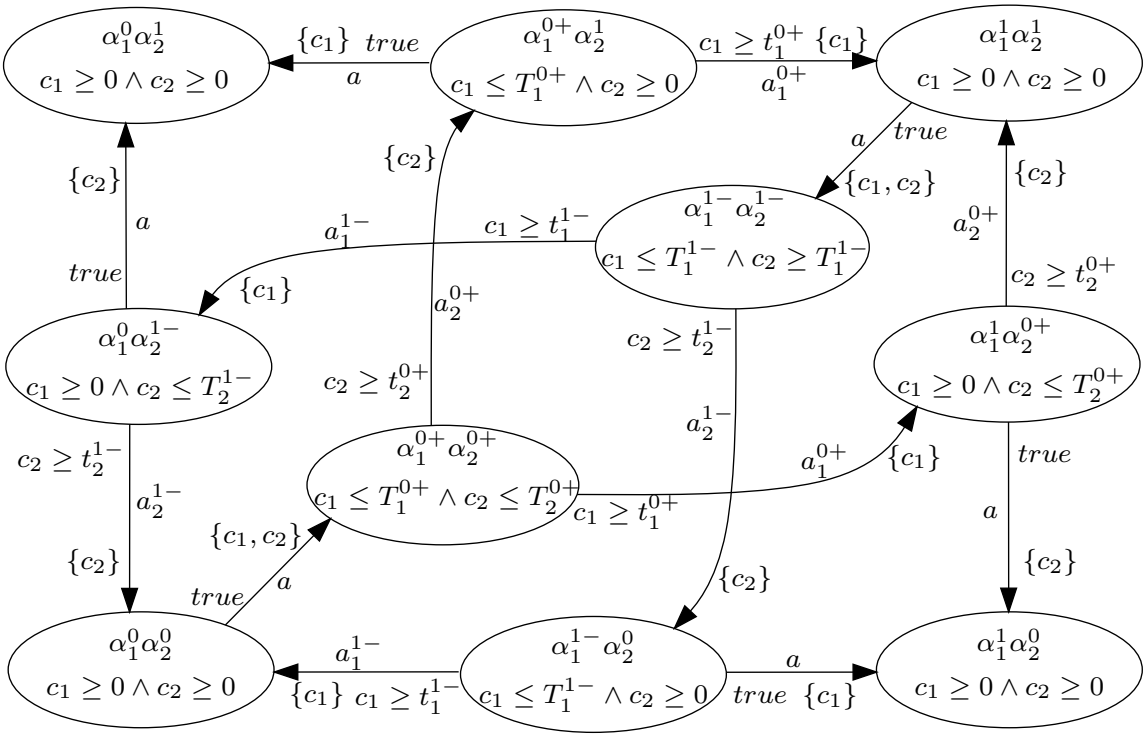



Figure 3: A part of the timed automaton $A$ constructed from $A_1$ and $A_2$ given in Fig. 2.

The event $a$ is used to indicate that the switch is defined by means of the switch conditions for the components $A_i$. Although the formal definition of the switches looks quite complicated, the actual meaning is straightforward. A location change occurs when the current state of locations allows for a change. The switch conditions $\Lambda_j^{k+}(l)$, $\Lambda_j^{k-}(l)$ carry the information which conditions, depending on the current location of $A_N$, the expression levels of the genes influencing $\alpha_j$ have to satisfy in order to induce a change in the expression level of $\alpha_j$ (see motivation of switch conditions of components $A_i$). Furthermore, changes in the expression level of a gene happen gradually. That is, for every two locations $l, l'$ connected by a switch we have $|\iota(l_i) - \iota(l'_i)| \leq 1$ for all $i \in \{1, \ldots, n\}$. The event $a$ is used to identify the switches that require checking the switch conditions of some location.

**Example 4.9.** *Fig. 3 shows a part of the automaton derived from the components $A_1$ and $A_2$ of our running example. Let us consider the location $(\alpha_1^0, \alpha_2^0)$. There is no switch in $A_1$ starting in location $\alpha_1^0$, and neither is there a switch in $A_2$ starting in $\alpha_2^0$ as can be seen in Fig. 2. So there is no switch starting in $(\alpha_1^0, \alpha_2^0)$ that originates from the switches of the components.*

*Let us now evaluate the switch conditions in $\alpha_1^0$ and $\alpha_2^0$ given in Fig. 2. We have $\iota(\alpha_1^0) = \iota(\alpha_2^0) = 0 < 1$. Thus both switch conditions are true in $(\alpha_1^0, \alpha_2^0)$. According to (2) in the preceding definition we obtain the switch $((\alpha_1^0, \alpha_2^0), a, true, \{c_1, c_2\}, (\alpha_1^{0+}, \alpha_2^{0+}))$. This switch represents the following situation. If both genes $\alpha_1$ and $\alpha_2$ have expression level zero, neither one inhibits the other. Therefore, both genes start to increase their expression level. This is represented by the state $(\alpha_1^{0+}, \alpha_2^{0+})$.*

*For both components of $(\alpha_1^{0+}, \alpha_2^{0+})$ there exist switches in $A_1$ resp. $A_2$. They are preserved*

*in the automaton $A$ and we obtain the switches $((\alpha_1^{0+}, \alpha_2^{0+}), a_1^{0+}, (c_1 \geq t_1^{0+}), \{c_1\}, (\alpha_1^1, \alpha_2^{0+}))$ and $((\alpha_1^{0+}, \alpha_2^{0+}), a_2^{0+}, (c_2 \geq t_2^{0+}), \{c_2\}, (\alpha_1^{0+}, \alpha_2^1))$. They represent the crossing of the corresponding threshold between expression level 0 and 1. Furthermore, we have to check the switch conditions in $\alpha_1^{0+}$ and $\alpha_2^{0+}$. Neither one is satisfied, so there is no switch starting in $(\alpha_1^{0+}, \alpha_2^{0+})$ labeled with $a$.*

*From state $(\alpha_1^1, \alpha_2^{0+})$ there is the switch $((\alpha_1^1, \alpha_2^{0+}), a_2^{0+}, (c_2 \geq t_2^{0+}), \{c_2\}, (\alpha_1^1, \alpha_2^1))$, which is preserved from $A_2$, as well as the switch $((\alpha_1^1, \alpha_2^{0+}), a, true, \{c_2\}, (\alpha_1^1, \alpha_2^0))$. The latter reflects the fact that the switch condition in $\alpha_2^{0+}$ is met, while the condition in $\alpha_1^1$ is not true in state $(\alpha_1^1, \alpha_2^{0+})$.*

Note that at this point we have not yet evaluated any of the time constraints placed on switches and locations. In the next step we have to determine which of the paths in the graph representing the automaton $A$ correspond to possible dynamical behaviors.

## 4.4 The Associated Transition System

In order to analyze the dynamics of the gene regulatory network represented by the timed automaton $A_N$ we consider the transition system associated with $A_N$ (see Def. 3.2). However, since we want to emphasize the importance of network structure and logical parameters for the dynamical behavior, we introduce a refined notion of transition system.

**Definition 4.10.** *Let $A_N := (L, L^0, \Sigma, C, I, E)$ be the network automaton of the network $N$, and let $T_{A_N} = (Q, Q^0, \Gamma, \rightarrow)$ be the transition system associated with $A_N$. Let $a \in \Sigma$ as given in Def. 4.8.*

- *Let $\rightarrow^{ur} := \{(q, \sigma, q') \in \rightarrow \mid (q, a, q') \in \rightarrow \ \Rightarrow \sigma \notin \mathbb{R}_{\geq 0}\}$. We call $T_{A_N}^{ur} := (Q, Q^0, \Gamma, \rightarrow^{ur})$ the transition system associated with $A_N$ and the* urgent event *$a$.*
- *Let $\rightarrow^{ov} := \{(q, \sigma, q') \in \rightarrow \mid (q, a, q') \in \rightarrow \ \Rightarrow \sigma = a\}$. We call $T_{A_N}^{ov} := (Q, Q^0, \Gamma, \rightarrow^{ov})$ the transition system associated with $A_N$ and the* overriding event *$a$.*

In the following, we call the event $a$ urgent or overriding to indicate which transition system will be analyzed. We drop the superscript *ur* resp. *ov* in the notation whenever we have clearly stated that we consider $a$ to be an urgent resp. overriding event.

The above definition leads to transition systems with a smaller set of possible transitions than the general transition system introduced in Def. 3.2. If $a$ is an urgent event, whenever there is some transition $(l, u) \xrightarrow{a} (l', v)$ for $(l, u), (l', v) \in Q$, we delete every transition of the form $(l, u) \xrightarrow{\delta} (l, u + \delta)$ regardless of the value of $\delta \in \mathbb{R}_{\geq 0}$. That is, whenever some transition is labeled with the urgent event $a$, it is not possible for time to elapse further in location $l$. However, there may be further discrete transitions starting in $(l, u)$.

**Example 4.11.** *To illustrate this, we assume $a$ to be urgent and take a look at location $(\alpha_1^{0+}, \alpha_2^1)$ in Fig 3. Assume we are in a state $(\alpha_1^{0+}, \alpha_2^1, \tau_1, \tau_2)$ with $\tau_1, \tau_2 \in \mathbb{R}_{\geq 0}$ and $\tau_1 < T_1^{0+}$. Since there is a switch labeled with $a$ starting in this location, we obtain two possible transitions starting in this state, namely $(\alpha_1^{0+}, \alpha_2^1, \tau_1, \tau_2) \xrightarrow{a} (\alpha_1^0, \alpha_2^1, 0, \tau_2)$ and $(\alpha_1^{0+}, \alpha_2^1, \tau_1, \tau_2) \xrightarrow{a_1^{0+}} (\alpha_1^1, \alpha_2^1, 0, \tau_2)$. Let $\delta \in \mathbb{R}_{\geq 0}$ such that $\tau_1 + \delta \leq T_1^{0+}$. Since $a$ is an urgent event, the transition $(\alpha_1^{0+}, \alpha_2^1, \tau_1, \tau_2) \xrightarrow{\delta} (\alpha_1^{0+}, \alpha_2^1, \tau_1 + \delta, \tau_2 + \delta)$ does not belong to $T_A^{ur}$.*

If we want to put even stronger emphasis on the switches derived from the switch conditions, we delete all transitions other than that labeled with $a$ starting in $(l, u)$, i. e., $a$ is an *overriding* event. Thus, in our example above only the switch mentioned first would remain. Unless otherwise stated, we assume in the following that $a$ is urgent.

Let us make one more observation about the transition system $T_{A_N}$ (and thus $T^{ur}_{A_N}$ and $T^{ov}_{A_N}$). Note that a transition labeled with $a$ never leads to a change in the expression levels of the genes, and that the set $J$ in the definition of switches of the network automaton is chosen maximal. Thus, if a path in $T_A$ starts in a regular location and its first transition is labeled with $a$, then the second transition in the path will not be labeled with $a$. This can be interpreted as follows. Starting in a regular state, all commands for a location change caused by the current distribution of resources, evaluated via the switch conditions, happen simultaneously. This leads to a location where each component is either in a location indicating increase or decrease of expression level or remains in the regular location of the original state. In this new state no switch condition of a component location is satisfied, since the expression levels of all components stayed the same. Thus, all transitions starting in the new location are either due to elapse of time or discrete transitions originating from some component $A_i$.

Again we are able to identify steady states of the system by the lack of outgoing edges. Here, a discrete state $l \in L$ is called a steady state if $T_A$ does not contain a discrete transition starting in $(l, u)$, for all clock interpretations $u$.

To analyze the dynamics of the gene regulatory network we consider the paths in $T_A$ that start in some initial state in $Q^0$. Questions of interest are for example if a steady state is reachable from a given initial location via some path in $T_A$. We will discuss the analysis of $T_A$ in a later section.

## 5 Comparison of the Models

In this section, we aim to show that on the one hand the information inherent in the state transition graph from Def. 2.5 can also be obtained from the transition system $T_A$ of a suitable timed automaton $A$. On the other hand, the modeling approach via timed automata offers possibilities to incorporate information about gene regulatory networks that cannot be included in the Thomas model, and thus leads to a refined view on the dynamics of the system.

Let $\mathcal{S}_N$ be the state transition graph corresponding to the gene regulatory network $N$ and $A = A_N = (L, L^0, \Sigma, C, I, E)$ the network automaton associated with $N$. We set $T_i^{k\varepsilon}, t_i^{k\varepsilon} = 0$ for all $i \in \{0, \dots, n\}$, $\varepsilon \in \{+, -\}$. Thus every guard condition evaluates to *true* and time does not elapse in the intermediate locations.

Next we derive a graph $G$ from $T_A = T^{ur}_A$. First we identify locations of $A_i$ representing the same expression level, i. e., for $k \in \{1, \dots, p_i - 1\}$ we define

$$v_k^{\alpha_i} := \{\alpha_i^k, \alpha_i^{k+}, \alpha_i^{k-}\}, \;\; v_0^{\alpha_i} := \{\alpha_i^0, \alpha_i^{0+}\} \text{ and } v_{p_i}^{\alpha_i} := \{\alpha_i^{p_i}, \alpha_i^{p_i-}\}.$$

Let $V^{\alpha_i} := \{v_k^{\alpha_i} \mid k \in \{0, \dots, p_i\}\}$ and $V := V^{\alpha_1} \times \cdots \times V^{\alpha_n}$ be the vertex set of $G$. Furthermore, there is an edge $v \to w$, if $v \neq w$ and if there is a path in $T_A$ from some state $(l, u)$, such that $l$ is regular, to a state $(l', u')$ satisfying $l'_i \in w_i$ for all $i$, such that every discrete state on the path other than $l'$ is an element of $v_1 \times \cdots \times v_n$. The condition to start in a regular state $l$ ensures that the first discrete transition occurring is labeled with $a$. This excludes the possibility of a

change of expression level that does not correspond to the parameter values. We can drop the condition, if we declare $a$ an overriding event.

Now, we need to show that $\mathcal{S}_N$ is contained in $G$. For the sake of completeness we prove the following stronger statement.

**Theorem 5.1.** *Suppose that the parameter constraints* (1) *given in Sect. 2.2 are satisfied. Then the graphs $\mathcal{S}_N$ and $G$ are isomorphic.*

*Proof.* We define $f : S \to V$, $(s_1, \ldots, s_n) \mapsto (v_{s_1}^{\alpha_1}, \ldots, v_{s_n}^{\alpha_n})$. Then it is easy to see that $f$ is a bijection.

Let $s \to s'$ be an edge in $\mathcal{S}_N$. We have to show that $f(s) \to f(s')$ is an edge in $G$. Set $v := f(s)$ and $w := f(s')$. According to the definition of edges in $\mathcal{S}_N$, there is a $j \in \{1, \ldots, n\}$ such that $s'_j = s_j + sgn(K_{\alpha_j, R_j(s)} - s_j) \neq s_j$ and $s_i = s'_i$ for all $i \in \{1, \ldots, n\} \setminus \{j\}$. Thus, $v_i = w_i$ for all $i \neq j$, and $v_j \neq w_j$.

First we consider the case that $s_j < K_{\alpha_j, R_j(s)}$. It follows that $s_j \neq p_j$, and thus $\alpha_j^{s_j}, \alpha_j^{s_j+} \in v_j$, and $s'_j = s_j + 1$. We choose $l \in L$ such that $l_i = \alpha_i^{s_i}$ for all $i \in \{1, \ldots, n\}$, thus $l \in v_1 \times \cdots \times v_n$ is regular. Furthermore, we choose the clock interpretation $u$ with $u(c) = 0$ for all $c \in C$.

We have $R_j(s) \subseteq Pred(\alpha_j)$ and, by definition, we know that $\lambda_j^{R_j(s)}(l)$, and thus the switch condition $\Lambda_j^{s_j+}(l)$, is true. It follows that there is a switch $(l, a, \varphi, R, \tilde{l}) \in E$ with $\varphi = true$, $\tilde{l}_j = \alpha_j^{s_j+}$ and $\tilde{l}_i \in v_i$ for all $i \neq j$. Thus we find a transition $(l, u) \xrightarrow{a} (\tilde{l}, u)$. Since time is not allowed to elapse in intermediate locations, and since no transition starting in $(\tilde{l}, u)$ is labeled with $a$ according to the observations made in the preceding section, every transition starting in $(\tilde{l}, u)$ will lead to a state that differs from $(\tilde{l}, u)$ in one component of the location vector only. Moreover, we have $(\alpha_j^{s_j+}, a_j^{s_j+}, \varphi_i^{s_j+}, \{c_j\}, \alpha_j^{s_j+1}) \in E_j$ and thus there is a transition $(\tilde{l}, u) \to (l', u)$ labeled with $a_j^{s_j+}$, with $l'_j = \alpha_j^{s_j+1} \in w_j$ and $l'_i = \tilde{l}_i \in v_i = w_i$ for $i \neq j$. It follows that $f(s) = v \to w = f(s')$ is an edge in $G$.

The case that $s_j > K_{\alpha_j, R_j(s)}$ and thus $s'_j = s_j - 1$ can be treated analogously.

Now let $v \to w$ be an edge in $G$. We set $s := f^{-1}(v)$ and $s' := f^{-1}(w)$. According to the definition there is a path $((l^1, u^1), \ldots, (l^m, u^m))$ in $T_A$ such that $l^1$ is regular, $l_i^j \in v_i$ for all $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m-1\}$ and $l_i^m \in w_i$ for all $i \in \{1, \ldots, n\}$. Since $l^1 \neq l^m$, there is some discrete transition in the path. Since every component of $l^1$ is regular, and thus the only discrete transition starting there is labeled by $a$, and since $a$ is an urgent event, we can deduce that $(l^1, u^1) \to (l^2, u^2)$ is labeled by $a$. Then $l^2$ has at least one component which is an intermediate location. Let $J \subseteq \{1, \ldots, n\}$ be such that $l_j^2$ is an intermediate location for all $j \in J$, and $l_i^2$ is a regular location for all $i \notin J$. Then $l_i^2 = l_i^1$ for all $i \notin J$. Since time is not allowed to elapse in the intermediate locations, the transition from $(l^2, u^2)$ to $(l^3, u^3)$ has to be discrete. Moreover, we know that the transition is not labeled by $a$, since the first transition of the path is already labeled that way. It follows that there is $j \in J$ such that $l_j^3$ is regular, $l_j^3 \neq l_j^2$, and $l_i^3 = l_i^2$ for all $i \neq j$. Furthermore, the expression levels of gene $\alpha_j$ in location $l_j^1$ and in location $l_j^3$ differ. We can deduce that $l_j^3 \notin v_j$ and thus $l_j^3 \in w_j$, $m = 3$ and $w_i = v_i$ for all $i \neq j$. We have $l_j^1 = \alpha_j^{s_j}$ and $l_j^3 = \alpha_j^{s'_j}$ and $|s_j - s'_j| = 1$.

We first consider the case that $s'_j = s_j + 1$, i. e., $l_j^1 = \alpha_j^{s_j}$, $l_j^2 = \alpha_j^{s_j+}$ and $l_j^3 = \alpha_j^{s_j+1}$. Since there is a transition from $(l^1, u^1)$ to $(l^2, u^2)$, we can deduce that the switch condition $\Lambda_j^{s_j+}(l^1)$ evaluates to *true*. Thus, there exists a subset $\omega$ of $Pred(\alpha_j)$ such that $K_{\alpha_j, \omega} > s_j$ and $\lambda_j^{\omega}(l^1)$

is true. By definition of the resources, we have $R_j(s) \supset \omega$ and thus $K_{\alpha_j,R_j(s)} \geq K_{\alpha_j,\omega} > s_j$ according to condition (1). It follows that $s'_j = s_j + 1 = s_j + sgn(K_{\alpha_j,R_j(s)} - s_j)$ and thus that $s \to s'$ is an edge in the state transition graph $\mathcal{S}_N$.

The case that $s'_j = s_j - 1$ can be treated analogously. □

In the above proof we used the most basic version of a timed automaton representing the network in question. Furthermore, we simplified the transition system $T_A$. Obviously, our modeling approach is designed to incorporate additional information about the biological system, e.g. about the actual values of synthesis and decay rates. Thereby we can obtain a more precise description of the dynamics of the system. For example, we may be able to discard certain paths in the state transition graph that violate conditions involving the time delays (see the example in the next section). Furthermore, we can evaluate stability and feasibility of a certain behavior, i. e., a path in the discrete transition system, in terms of clock interpretations that allow for that behavior. The stricter the conditions the clock interpretations have to satisfy to permit a certain behavior, the less allowance is made for fluctuations in the actual time delays of the genes involved.

The intermediate locations give supplementary information about the behavior of the genes. For instance, it is possible to distinguish between a gene keeping the same expression level because there is no change in the expression levels of the genes influencing it, and the same behavior due to alternating opposed influences. In the first case, the gene stays in the regular location representing the expression level, in the latter case it also traverses the corresponding intermediate locations.

Moreover, although this model uses asynchronous updates, it also allows for synchronous updates in the sense that two discrete transitions may occur at the same point in time. That is due to the fact that transitions labeled with $a$ never result in a change of expression level. But, on the other hand those transitions are the only ones resulting in a change in more than one component of the location vector. Thus a change of expression level for more than one component cannot happen in a single transition. However, it is possible that two transitions resulting in change of expression level are executed successively, i. e., time does not elapse between those transitions. This may lead to paths in the transition system that are not incorporated in the state transition graph of the Thomas formalism.

To clarify the above considerations we give an illustrative example in the next section.

# 6 Bacteriophage $\lambda$

Temperate bacteriophages are viruses that can act in two different ways upon infection of a bacterium. If they display the lytic response, the virus multiplies, kills and lyses the cell. However, in some cases the viral DNA integrates into the bacterial chromosome, rendering the viral genome harmless for the so-called lysogenic bacterium.

## 6.1 Two Genes Model

In [12], the formalism of Thomas is used to describe and analyze the genetic network associated with this behavior. Fig. 1 shows the simplified model there, which we already used as a running example in the preceding sections. The gene $\alpha_1$ corresponds to the gene *cI* and $\alpha_2$ to the gene *cro* of the bacteriophage $\lambda$. The choice of the thresholds and parameter values is based

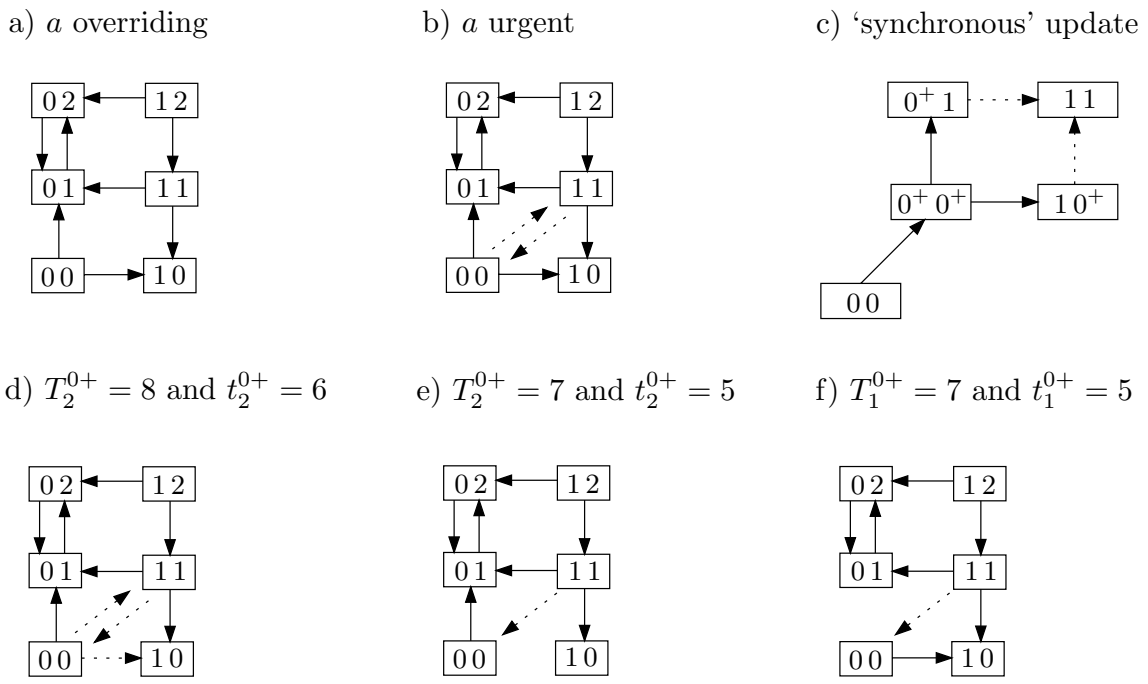



Figure 4: Graphs representing the dynamical behavior of the system derived from the transition systems resulting from different specifications of the model. Unless otherwise stated $a$ is an urgent event and we set $T_i^{k+} = T_i^{k-} = 10$ and $t_i^{k+} = t_i^{k-} = 8$ for all $i \in \{1, 2\}$ and $k \in \{0, 1, 2\}$.

on experimental data. As already mentioned, they render the loop starting in $\alpha_1$ ineffective with respect to the dynamics. The resulting state transition graph in Fig. 1 shows two possible behaviors. The steady state in $(1, 0)$ can be related to the lysogenic, the cycle comprising the states $(0, 1)$ and $(0, 2)$ to the lytic behavior.

Now we analyze this network modeled as a timed automaton $A$ with components $A_1$ and $A_2$, see Fig. 2. A part of $A$ is shown in Fig. 3, but we have not yet considered the corresponding transition system. Fig. 4 displays graphs, which are condensed versions of the different transition systems derived from $A$. With the exception of graph (c), the vertices of the graphs represent the expression levels of the genes, which correspond to the integer value of the location superscript. For instance, states $(\alpha_1^0, \alpha_2^{1-})$ and $(\alpha_1^{0+}, \alpha_2^1)$ are both represented by $(0, 1)$.

We analyze the dynamics of the system starting only from regular states. Thus, edges as well as paths in the graphs from a vertex $(j_1, j_2)$ to a vertex $(i_1, i_2)$ signify that the system can evolve from $(\alpha_1^{j_1}, \alpha_2^{j_2})$ to a state where $\alpha_1$ and $\alpha_2$ have expression level $i_1$ and $i_2$ respectively. Thereby it traverses states with expression levels corresponding to the vertices in the path, provided there is an actual point in time in which the genes acquire those expression levels. Again graph (c) is an exception to this representation and its analysis will clarify the distinction.

We specify our model by choosing values for the maximal and minimal time delays. Set $T_i^{k+} = T_i^{k-} = 10$ and $t_i^{k+} = t_i^{k-} = 8$ for all $i \in \{1, 2\}$ and $k \in \{0, 1, 2\}$. That is to say, the time delays for synthesis and decay are all in the same range regardless of the gene and the expression level. If we declare $a$ to be an overriding event, we avoid the possibility that there is a path from $(0, 0)$ to $(1, 1)$ in the graph derived from the corresponding transition system as

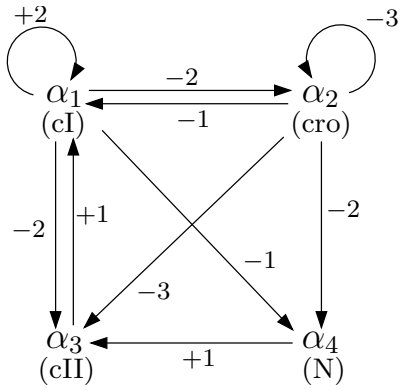
explained in Sect. 4.4. This is illustrated in Fig. 4 (a) and matches the state transition graph in Fig. 1. In (b), $a$ is again an urgent event. We obtain two opposite edges between $(0,0)$ and $(1,1)$. However, there are very strict conditions imposed on the time delays in order for the system to traverse those edges, which we drew dotted for that reason. To clarify the situation, we follow the path from $(0,0)$ to $(1,1)$ via the intermediate states shown in (c). A switch labeled with $a$ leads to $(0^+,0^+)$. Assuming that $\alpha_1$ reaches the next expression level faster than $\alpha_2$ after a time delay $8 \leq r_1 \leq 10$, we reach $(1,0^+)$. In that situation two switches are enabled. One is labeled by $a$ and leads to $(1,0)$. Since time is not allowed to pass, whenever the actual time $r_2$ that $\alpha_2$ needs to reach the expression level 1 differs from $r_1$, that switch is taken. Only in the case that both time delays are exactly equal, the system will move via the switch labeled by $a_2^{0+}$ to $(1,1)$. Analogous considerations apply to the path via $(0^+,1)$. It follows that although states $(0,0)$ and $(1,1)$ form a cycle in the graph, it is not plausible that the system will traverse that cycle. Once in the cycle, even the slightest perturbation of one of the time delays suffices for the system to leave the cycle. It is unstable.

These considerations apply not only to the edges representing synchronous update. In Fig. 4 (d) we change the values for $T_2^{0+}$ and $t_2^{0+}$ to express that the synthesis of $\alpha_2$ is usually faster than that of $\alpha_1$. The system can reach the state $(1,0)$ only if $\alpha_2$ needs the maximal and $\alpha_1$ the minimal time to change their expression level. So, usually we would expect the system to reach the cycle comprising $(0,1)$ and $(0,2)$, corresponding to the lytic behavior of the bacteriophage. If we know that $\alpha_2$ is always faster than $\alpha_1$ in reaching the expression level 1, we can altogether eliminate both the edge leading from $(0,0)$ to $(1,0)$, and the one leading to $(1,1)$, as shown in (e). There is no clock interpretation satisfying the imposed conditions. If we reverse the situation of $\alpha_1$ and $\alpha_2$, we eliminate the edges from $(0,0)$ to $(0,1)$ and $(1,1)$ as shown in (f). In this case, the system starting in $(0,0)$ will always reach the steady state $(1,0)$ representing the lysogenic response of the bacteriophage.

We have implemented the above system in UPPAAL, a tool for analyzing systems modeled as networks of timed automata ([3], and `http://www.uppaal.com`). Since UPPAAL uses product automata in the sense of [1], we had to make some modifications in the modeling of the components. Primarily, we converted the switch conditions to actual switches, which synchronize via the input of an external component that ensures the desired update mechanisms of the system. Using the UPPAAL model checking engine, we verified the above mentioned dynamical properties of the different specifications of our model.

## 6.2 Four Genes Model

In [12] the authors furthermore discuss a more complete model of the considered bacteriophage system. They include the effects of not only genes *cI* and *cro* but also of genes *cII* and *N*. The resulting model is shown in Fig. 5, the parameter values are derived from experimental data and theoretical considerations (see [12] for details). Again the lytic and lysogenic behavior can be identified in the state transition graph. The former is represented by the steady state $(2,0,0,0)$, the latter by the cycle $C$ comprising the states $(0,2,0,0)$ and $(0,3,0,0)$. The timed automaton model consists of four components. Component $A_1$ corresponding to *cI* includes three regular and four intermediate locations. Component $A_2$ corresponding to *cro* includes four regular and six intermediate locations, and components $A_3$ and $A_4$ corresponding to *cII* and *N* both have two regular and two intermediate locations. The parameter values satisfy condition (1) which simplifies the switch conditions.




$$
\begin{array}{lcllcl}
K_{\alpha_1,\{\alpha_2\}} & = & 2 & K_{\alpha_2,\{\alpha_1\}} & = & 2 \\
K_{\alpha_1,\{\alpha_3\}} & = & 2 & K_{\alpha_2,\{\alpha_1,\alpha_2\}} & = & 3 \\
K_{\alpha_1,\{\alpha_1,\alpha_2\}} & = & 2 & & & \\
K_{\alpha_1,\{\alpha_1,\alpha_3\}} & = & 2 & K_{\alpha_3,\{\alpha_1,\alpha_2,\alpha_4\}} & = & 1 \\
K_{\alpha_1,\{\alpha_2,\alpha_3\}} & = & 2 & & & \\
K_{\alpha_1,\{\alpha_1,\alpha_2,\alpha_3\}} & = & 2 & K_{\alpha_4,\{\alpha_1,\alpha_2\}} & = & 1
\end{array}
$$

Figure 5: Extended model of the bacteriophage $\lambda$ network. Only non-zero parameter values are given

Again we have implemented the model in UPPAAL and analyzed the dynamical behavior for different time constraints, starting with identical time values regardless of corresponding gene and expression level. As a first result, we note that the crucial roles of *cI* and *cro*, i. e., $\alpha_1$ and $\alpha_2$, are preserved in the following sense. If the values of $T_1^{k+}$ and $t_1^{k+}$ for $k \in \{0,1,2\}$ are sufficiently big in comparison with $T_2^{l+}$ and $t_2^{l+}$ for $l \in \{0,1,2,3\}$, then the state $(2,0,0,0)$ is not reachable from the initial state $(0,0,0,0)$ while the cycle $C$ is reachable. That is, if the expression level increase of *cI* is slow relative to that of *cro*, the system shows lytic behavior. If the situation is reversed, the system shows lysogenic behavior. These observations are in accordance with the results for the two gene model as illustrated in Fig. 4.

Now, we want to give an example how to incorporate information on the system gathered from biological experiments (see [7] for an overview). When transcription of the bacteriophage DNA is initiated, *cro* and *N* are the first of our model components that are expressed. This is exactly the kind of information we need in order to obtain a more concise model. Starting in the state $(0,0,0,0)$ the given parameter values indicate expression level increase for the genes *cI*, *cro* and *N*. According to the given data, we choose the values $T_2^{0+}$, $T_4^{0+}$, $t_2^{0+}$ and $t_4^{0+}$ smaller than the values $T_1^{0+}$ and $t_1^{0+}$. The impact on the dynamical behavior of the system is illustrated in Fig. 6 (a) and (b). In (a) we see part of the state transition graph obtained from the model given in Fig. 5. Starting in the state (0,0,0,0), representing the inactivity of the bacteriophage DNA when introduced into the bacterial cell, all possible changes in expression level, i. e., for genes *cI*, *cro* and *N*, are taken into account. In the state transition graph of the Thomas formalism we thus have three outgoing edges. From the corresponding target states again every possible change in expression level is included in the graph, resulting in a strongly branched structure. The impact of the additional information on the time constraints given above is shown in (b). The transition from initial state $(0,0,0,0)$ to $(1,0,0,0)$ can be excluded since it violates the temporal constraints. Furthermore, we can derive that the second transition has to lead to state $(0,1,0,1)$, representing expression of the genes *cro* and *N* in accordance with the experimental observations. Thus, we are able to exclude sizeable parts of the state transition graph.

We want to point out a further advantage of our approach by taking a closer look to the dynamics presented in Fig. 6. The part of the state transition graph given in (a) contains two cycles. First, we have the cycle $C$ representing the lytic behavior, second the cycle $C'$

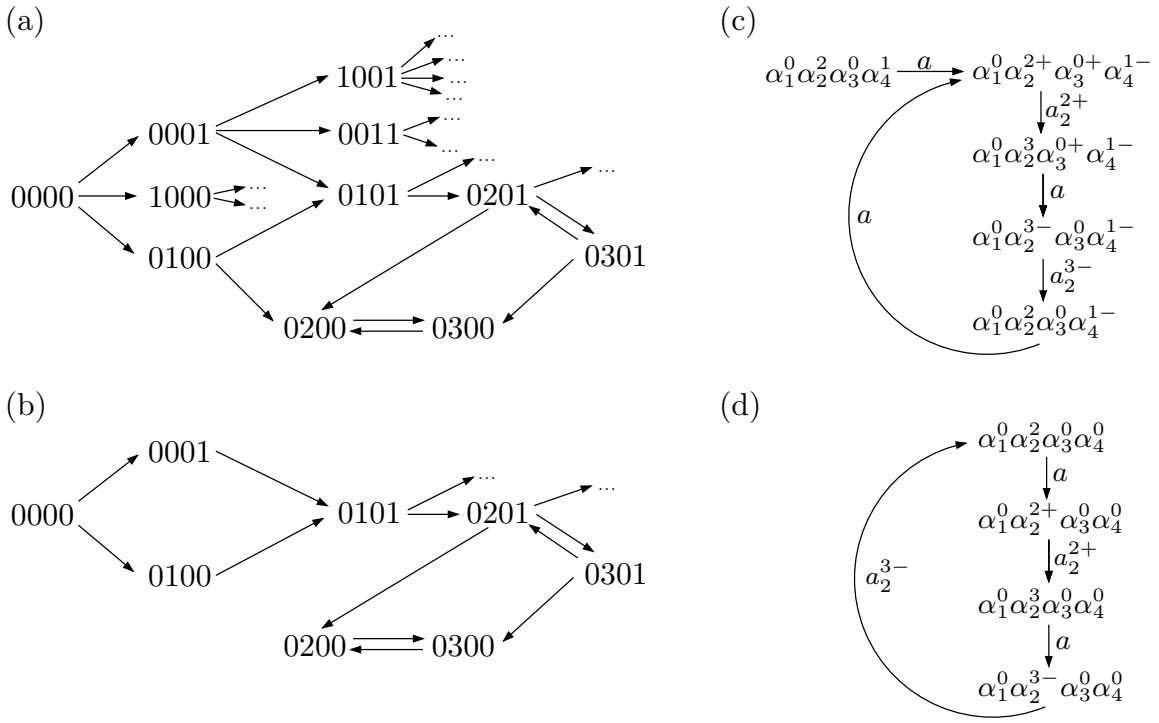



Figure 6: Parts of the state transition graph resp. transition systems of the model given in Fig. 5

comprising the states $(0, 2, 0, 1)$ and $(0, 3, 0, 1)$. According to the information inherent in the state transition graph, the system could remain in the cycle $C'$ indefinitely. Fig. 6 (c) shows the cycle in a suitable transition system derived from the timed automaton model. Starting from the regular location $(\alpha_1^0, \alpha_2^2, \alpha_3^0, \alpha_4^1)$ we enter the cycle by evaluating the switch conditions and executing a switch labeled with $a$. Now, $\alpha_2$ has to reach its next expression level sooner than $\alpha_3$ and $\alpha_4$ in order to reach the state $(\alpha_1^0, \alpha_2^3, \alpha_3^{0+}, \alpha_4^{1-})$, corresponding to the state $(0, 3, 0, 1)$. Continuing in that manner, we derive time constraints that have to be satisfied if the system is to remain in the cycle. However, a close look shows that $A_4$ always remains in the location $\alpha_4^{1-}$ representing the process of gene $N$ decreasing its expression level from one to zero. After the corresponding time delay, $N$ will reach the expression level zero, thus forcing the system out of the cycle. Cycles with this property have already been considered in [14].

The cycle $C$ representing the lytic behavior, as illustrated in Fig. 6 (d), shows opposite characteristics. We enter the cycle from the regular state $(\alpha_1^0, \alpha_2^2, \alpha_3^0, \alpha_4^0)$. The given parameter values, evaluated by means of the switch conditions, indicate a change in expression level for gene *cro* only, leading to a transition to $(\alpha_1^0, \alpha_2^{2+}, \alpha_3^0, \alpha_4^1)$. In fact, all transitions in the cycle concern solely gene *cro*. The other genes remain in their respective stable regular location. Thus, we can deduce that the actual values of the time delays concerning the expression level changes from two to three and back do not influence the behavior of the system after reaching the cycle $C$.

Lastly, we take another look at the key players in the network. While the genes *cI* and *cro*

define the lysogenic and lytic states respectively, studies have shown the importance of *cII* in the switching process (see [7]). Since *cro* and *N* are the first to be expressed, the activation of *cI* transcription heavily depends on *cII*. As for the time constraints concerning *cI* and *cro* discussed above, we find that we can determine the decision between lysogeny and lytic behavior by evaluating the *cII* time delays with respect to those of *cro*. Again, if the expression level increase of *cII* is sufficiently slow in comparison with that of *cro*, the system will reach the lytic state. In the reversed situation, we obtain lysogeny. However, the impact of *cII* cannot be correctly captured by our model. The reason for this is the following. The parameter values for *cI* reflect that *cI* can reach its highest expression level in the absence of *cro*, as well as in the presence of *cII*. Thus, if either condition is met, the system moves to the location representing increase of expression level which is labeled with the corresponding time delay. However, the rate for synthesis of *cI* in the presence of *cII* is much higher than in its absence. Thus, we need to consider distinct time delays which depend on the current expression level of *cII*, if we want to capture the resulting impact on the dynamical behavior. This issue will be addressed again in the last section.

# 7 Perspectives

In this paper, we introduced a discrete modeling approach that extends the established formalism of Thomas by incorporating constraints on the time delays occurring in the operations of biological systems. We addressed some of the advantages this kind of model offers, but naturally there is much room for future work. One of the most interesting possibilities the model provides is the evaluation of feasibility and stability of certain behaviors of the system by means of the constraints imposed on the time delays. We may find cycles in the transition system (implying homeostatic behavior of the real system), the persistence of which requires that equalities for time delays are satisfied. It is highly unlikely that a biological system will sustain a behavior which does not allow for the slightest perturbation in its temporal processes. A cycle persisting for a range of values for each time delay will be a lot more stable. The merit of such considerations was already mentioned by Thomas (see [15]). It calls for a thorough analysis using mathematical methods as well as testing with substantial biological examples.

Furthermore, it seems worthwhile to relax some of the conditions imposed by the Thomas formalism. We already emphasized the local character of our approach regarding the interactions between the components of the network. Thus we were able to avoid the restrictions induced by the parameter constraints (1). It also could be advantageous to allow a gene product to influence a target gene depending on its concentration. For instance, it may be activating at low but inhibiting at high concentrations. Our approach clearly allows for the modeling of such a situation, however the underlying formal framework, i. e., the definitions of interaction graph, resources and/or parameter values, has to be adapted to obtain a concise description.

A related observation is that our modeling formalism does not allow one to distinguish between processes of expression level change, represented by some intermediate location, caused by different situations. That is, we cannot capture the behavior of a system where the change of expression level from some level $k$ to $k+1$ (or $k-1$) occurs with different time delays depending on the state of the system. This could be achieved by allowing for different intermediate locations that represent the same process of expression level change. However, some thought has to be given to the possible definition of switches between such intermediate locations.

We would like to close with some remarks regarding the analysis of the dynamics of our model. The theory of timed automata provides powerful results concerning analysis and verification of the model by means of model checking techniques. For example, CTL and LTL model checking problems can be decided for timed automata (see [2]). However, we face the state explosion problem and moreover the task to phrase biological questions in terms suitable for model checking. A thorough study of problems and possibilities of applying model checking techniques to answer biologically relevant questions using the modeling framework given in this paper seems necessary and profitable.

## References

[1] R. Alur. Timed Automata. In *Proceedings of the 11th International Conference on Computer Aided Verification*, volume 1633 of *LNCS*, pages 8–22. Springer, 1999.

[2] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas. Discrete abstractions of hybrid systems. In *Proceedings of the IEEE*, 2000.

[3] J. Bengtsson and W. Yi. Timed Automata: Semantics, Algorithms and Tools. In *Lecture Notes on Concurrency and Petri Nets*, volume 3098 of *LNCS*, pages 87–124. Springer, 2004.

[4] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. Application of formal methods to biological regulatory networks: extending Thomas' asynchronous logical approach with temporal logic. *J. Theoret. Biol.*, 229:339–347, 2004.

[5] L. Glass and S. A. Kauffman. The logical analysis of continuous, non-linear biochemical control networks. *J. Theoret. Biol.*, 39:103–129, 1973.

[6] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theoret. Biol.*, 22:437–467, 1969.

[7] A. Oppenheim, O. Kobiler, J. Stavans, D. Court, and S. Adhya. Switches in bacteriophage lambda development. *Annu. Rev. Genet.*, 39:409–429, 2005.

[8] E. Remy, P. Ruet, and D. Thieffry. Graphic requirements for multistability and attractive cycles in a boolean dynamical framework. Prépublication, 2005.

[9] H. Siebert and A. Bockmayr. Incorporating time delays into the logical analysis of gene regulatory networks. In *Computational Methods in Systems Biology, CMSB 2006, Trento, Italy*, volume 4210 of *LNCS*, pages 169–183. Springer, 2006.

[10] E. H. Snoussi and R. Thomas. Logical identification of all steady states: the concept of feedback loop characteristic states. *Bull. Math. Biol.*, 55:973–991, 1993.

[11] M. Sugita. Functional analysis of chemical systems in vivo using a logical circuit equivalent. *J. Theoret. Biol.*, 1:415–430, 1961.

[12] D. Thieffry and R. Thomas. Dynamical behaviour of biological regulatory networks - II. Immunity control in bacteriophage lambda. *Bull. Math. Biol.*, 57:277–297, 1995.

[13] R. Thomas. Boolean formalisation of genetic control circuits. *J. Theoret. Biol.*, 42:565–583, 1973.

[14] R. Thomas and R. d'Ari. *Biological Feedback*. CRC Press, 1990.

[15] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. II. Logical analysis of regulatory networks in terms of feedback circuits. *Chaos*, 11:180–195, 2001.

# Chapter 8

# Stochasticity in Reactions: A Probabilistic Boolean Modeling Approach

Sven Twardziok, Heike Siebert and Alexander Heyl.

In Proceedings of the 8th International Conference on *Computational Methods in Systems Biology*, CMSB 2010, Trento, Italy, pages 76-85, ACM, 2010.

**Remarks.** This paper elaborates some results Sven Twardziok developed in his master thesis and a preceding research project, for which I was one of the supervisors. The aim was to elucidate a certain feedback mechanism in the cytokinin signaling network of *Arabidopsis thaliana*. Alexander Heyl, a plant biologist from Freie Universität Berlin, provided data and expertise for the modeling of the network. We started with a discrete model, but it became evident that the framework was to coarse to allow for a meaningful analysis. Sven Twardziok developed a new modeling approach integrating stochastic aspects in a Boolean framework, tailored to the needs recognized while collaborating with the biologists on the model building. In addition, he provided ideas and methods for the analysis of partially specified models, i.e., models where initially some of the parameters are not determined. I coordinated the research process and clarified and extended the proposed ideas, particularly those concerning the comparison with existing stochastic Boolean frameworks and the analysis approach. The final version of the paper was written jointly by all authors.

# Stochasticity in Reactions: A Probabilistic Boolean Modeling Approach

Sven Twardziok, Heike Siebert and Alexander Heyl

*Abstract.* Boolean modeling frameworks have long since proved their worth for capturing and analyzing essential characteristics of complex systems. Hybrid approaches aim at exploiting the advantages of Boolean formalisms while refining expressiveness. In this paper, we present a formalism that augments Boolean models with stochastic aspects. More specifically, biological reactions effecting a system in a given state are associated with probabilities, resulting in dynamical behavior represented as a Markov chain. Using this approach, we model and analyze the cytokinin response network of *Arabidopsis thaliana* with a focus on clarifying the character of an important feedback mechanism.

## 1 Introduction

Today, mathematical modeling is an integral part of systems biology research. Mathematical formalisms not only offer a rigorous way of collecting information on a given system, but also allow for comprehensive analysis of structural aspects and dynamical behavior of biological networks, testing of hypotheses and a focused approach to experimental design. As a first step in mathematical modeling, we need to choose a modeling formalism capable of representing the system. If enough data of sufficient precision is available, differential equation modeling is often well suited to the task. If quantitative information is lacking and kinetic mechanisms are unknown, or if low copy numbers of biological entities have to be modeled, discrete modeling methods are a good choice.

The decision which modeling formalism to choose is often not clear-cut. Discrete methods might not be able to capture some important aspects of a biological system, a fully quantitative model might be out of reach. Hybrid systems allow to keep the simplicity of discrete models while expanding the expressiveness of the formalism for some aspects. For example, a continuous time flow might be added to a discrete state space in order to evaluate the effects of different time delays on the system's behavior (see e. g. [3, 11]). In this paper we introduce a Boolean modeling framework extended to include stochastic aspects. In the following, we shortly recall basic concepts concerning Boolean modeling and stochastic extensions of Boolean frameworks. In Sect. 2, we introduce the *Stochasticity in Reactions* framework, which is based on a local Boolean modeling approach and introduces probabilities for the processes generating the network dynamics. We shortly discuss several analysis approaches, before we illustrate the method by modeling and analyzing the cytokinin signal transduction network in *Arabidopsis thaliana* in Sect. 3. Joint work of plant biologists, mathematicians and computer scientists on this signaling network, with a focus on identifying the mechanism behind an inhibitory feedback effect involved

in regulation of the signal transduction, led to the formalism as it is presented in this paper. We close with a short discussion and perspectives for future work.

### 1.1 Boolean Models

A Boolean model is a very abstract representation of a system. Assume we consider a system consisting of $N \in \mathbb{N}$ components. Each component is represented by a Boolean variable. Interpretation of the variable values differs considerably depending on the nature of the components modeled. A component having value one might signify that the concentration of the corresponding substance exceeds a given threshold, it might indicate that a gene is being expressed or that a receptor receives a signal and so on. In any case, the state space of the system is the set $\mathcal{S} := \{0,1\}^N$.

The dynamics of the system is encoded in a Boolean function $f : \mathcal{S} \to \mathcal{S}$, whose coordinate functions (called *state variable update functions*) $f_i$, $i \in \{1,\dots,N\}$, determine the behavior of the corresponding components. Given the function $f$ and the state space $\mathcal{S}$, a state transition graph $G := (V,E)$ representing the system's dynamics can be derived. The vertex set of this graph corresponds to the state space of the system and edges are defined based on the function $f$ and on a chosen update strategy. The most common update strategies are the so-called synchronous update linking a state with its image under $f$ and the asynchronous update that does not allow for simultaneous update of component values and results in a non-deterministic representation of the dynamics (see e. g. [7, 12] for definitions).

Boolean models often do not capture all important aspects of a given system, however, they still allow for rigorous representation and analysis and often reveal fundamental properties of the system. Hybrid methods are a way to refine the Boolean approach and to incorporate effects that cannot be modeled in the Boolean framework, such as time delays or stochastic aspects.

### 1.2 Stochastic expansion

Quite some work has been done considering stochastic expansions of Boolean models, motivated by the non-deterministic nature of many biological systems. Substance concentrations might fluctuate over time and depending on environmental conditions, perturbations or damage might occur and so on. Different approaches incorporating stochastic aspects in Boolean frameworks can be distinguished based on the way stochasticity is included. Some examples for such methods are Probabilistic Boolean Networks [10], Stochasticity in Nodes [9], Stochasticity in Functions [4] and Probabilistic Asynchronous Update [16]. These approaches are closely related but emphasize different aspects of modeling and analyzing biological systems. In the following, we focus exemplarily on the Probabilistic Boolean Network approach for illustrative and comparative purposes.

In a probabilistic Boolean model, the stochastic events effecting a system in a state $s \in \mathcal{S}$ are represented as elementary events of a finite *probability space* $\Omega_s := \{\omega_1,\dots,\omega_k\}$, $k \in \mathbb{N}$. A probability function $P_s : \Omega_s \to [0,1]$ assigns probabilities to the events. By definition the sum of the probabilities of the elementary events $\omega \in \Omega_s$ is one. In a slight abuse of notation we drop the index $s$ and denote by $P$ the probability function for $\Omega_s$ for all $s \in \mathcal{S}$. Evolution of the system in state $s$ is then determined by a random experiment in $\Omega_s$. We denote the set of pairs of a state $s \in \mathcal{S}$ and an elementary event $\omega \in \Omega_s$ by $\mathcal{O}$:

$$\mathcal{O} := \{(s,\omega) : s \in \mathcal{S}, \omega \in \Omega_s\}.$$

The *probabilistic transition function* $\mathsf{f} : \mathcal{O} \to \mathcal{S}$ calculates the successor state for a state $s$ and an elementary event $\omega \in \Omega_s$. Each elementary event $\omega \in \Omega_s$ with $\mathsf{f}(s, \omega) = s'$ represents a possible state transition from $s$ to $s'$. We illustrate this idea in Figure 1. The system starts in a state $s$ with probability space $\Omega_s$. The next state $s'$ is chosen by a random experiment in $\Omega_s$ in agreement with $\mathsf{f}(s, \omega)$.

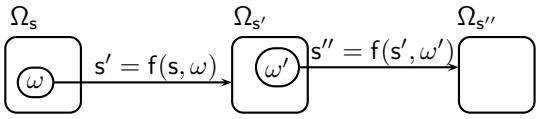



Figure 1: State transitions of a probabilistic Boolean model.

The dynamics of a probabilistic Boolean model is described by a *probabilistic state transition graph*, i. e. a directed weighted graph with a state space as its vertex set. Edges represent possible state transitions while edge weights represent transition probabilities.

**Definition 1.1.** *We call the graph $\mathcal{G} = (V, \rho, E)$ with vertex set $V := \mathcal{S}$, a function*

$$\rho : \mathcal{S} \times \mathcal{S} \to [0, 1], \quad \rho(s, s') = \sum_{\substack{\omega \in \Omega_s, \\ \mathsf{f}(s,\omega) = s'}} P(\omega),$$

*and edge set $E := \{(s, s') \in V \times V \,:\, \rho(s, s') > 0\}$ a* probabilistic state transition graph.

### Probabilistic Boolean Networks

In a Probabilistic Boolean Network (PBN), there exist $l(n) \in \mathbb{N}$ alternative state variable update functions $f_n^1, \ldots, f_n^{l(n)}$ for each state variable $n \in \{1, \ldots, N\}$. A function $f_n^{k_n}$ with $k_n \in \{1, \ldots, l(n)\}$ is chosen for the next update of state variable $n$ with a predefined probability $c_n^{k_n} \in [0, 1]$. Naturally, the sum of the update function probabilities associated with a state variable $n$ must be equal to one:

$$\sum_{k=1}^{l(n)} c_n^k = 1\,.$$

The choices of state variable update functions are independent of each other as well as of the current state of the system. A state transition is derived using synchronous update, i. e., all state variables are updated simultaneously according to the chosen update functions. The probability space for an arbitrary state $s \in \mathcal{S}$, is then defined as

$$\Omega_s = \Big\{(k_1, \ldots, k_N) : k_n \in \{1, \ldots, l(n)\}, n \in \{1, \ldots, N\}\Big\}\,.$$

It does not depend on the choice of $s$. The probability for a single elementary event $\omega = (k_1, \ldots, k_N)$ is given by

$$P(\omega) = \prod_{n=1}^{N} c_n^{k_n}\,.$$

The probabilistic transition function is then defined by

$$\mathsf{f}(s, \omega) = \big(f_1^{k_1}(s), \ldots, f_N^{k_N}(s)\big)$$

for all $s \in \mathcal{S}$ and $\omega = (k_1, \ldots, k_N) \in \Omega_s$.

# 2 Stochasticity in Reactions

## 2.1 Model description

The Boolean modeling formalism underlying our framework is based on a description of the biological processes making up the system's behavior as a whole. Rather than giving a global description of the dynamics via a function $f$, this approach allows us to model in a local fashion often more suited to translate the available knowledge into a model. Extending the Boolean approach, we associate probabilities with the different reactions possible between network components. Referring to the approaches [9, 4] where stochasticity is introduced by considering different global update functions, we call our approach *Stochasticity in Reactions* (SIR).

As a first modeling step, we describe possible reactions in the system qualitatively. Here, a reaction has a local effect on the system in the sense that it influences a subset of system components. For example, we can model substance degradation as a reaction that only influences the substance in question, while a biochemical reaction consuming and producing some substances will affect every network component representing one of the substances involved. To describe a reaction we represent its effect on the state of the system as a vector. Furthermore, we assign a probability for the reaction to occur which naturally depends on the state of the system.

**Definition 2.1.** *A* reaction *is a pair $r := (e, p)$ consisting of an effect vector $e \in \{-1, 0, 1\}^N$ and a function $p : S \to [0, 1]$ satisfying $p(s) = 0$ for all $s \in \mathcal{S}$ with $s + e \notin \mathcal{S}$. We say that $r$ is* valid in a state $s \in S$ *if $p(s) \neq 0$.*

The vector $e$ specifies the way certain state variables are affected by the reaction. The components with value zero represent network components unaffected by the reaction. Note that different reactions may have the same effect vector. The function $p$ calculates the probability of the reaction taking place in a given state. In the following, we represent $p$ as $p = b^e(s) \cdot pr$, where $b^e : \mathcal{S} \to \{0, 1\}$ is a Boolean function and $pr \in [0, 1]$ is a probability value. Thus, the probability for a reaction is either zero or $pr$. The Boolean function $b^e$ indicates whether the execution of a reaction in a given state is possible at all. That is, while the effect vector encodes the impact of a reaction, the function $b^e$ encodes the conditions necessary for the reaction to occur. The probability value then is the same for all states that satisfy those conditions. This modeling assumption has proved sufficient for our purposes so far. Nevertheless, it is certainly worth thinking about modeling the probability with greater dependence on the current state, i. e., considering a function $pr : \mathcal{S} \to [0, 1]$ instead of a constant value $pr$.

Given a set of reactions $\mathcal{R}$, the set of all valid reactions in a state $s$ is denoted with $R_s := \{r \in \mathcal{R} : p(s) \neq 0\}$.

The following example will be used to illustrate the formalism during the first part of our paper.

**Example 2.2.** *Consider a toy network with three state variables, i. e. $\mathcal{S} = \{0, 1\}^3$, and a set of four reactions $R := \{r_1, r_2, r_3, r_4\}$. Each reaction is defined by its effect vector and probability function:*

$$
\begin{aligned}
r_1 &: \; e_1 = (1, -1, 0), \; p_1(s_1, s_2, s_3) = (1 - s_1) \cdot s_2 \cdot pr_1 \,, \\
r_2 &: \; e_2 = (0, 1, -1), \; p_2(s_1, s_2, s_3) = (1 - s_2) \cdot s_3 \cdot pr_2 \,, \\
r_3 &: \; e_3 = (-1, 0, 0), \; p_3(s_1, s_2, s_3) = s_1 \cdot pr_3 \,, \\
r_4 &: \; e_4 = (1, 0, 1), \; p_4(s_1, s_2, s_3) = (1 - s_1) \cdot (1 - s_3) \cdot pr_4 \,.
\end{aligned}
$$

*Here, $r_1$ could represent a transformation of a substance represented by the second network component to a substance represented by the first component, which is encoded in the first two components of the effect vector. We want to model that the transformation can only occur, if there is no substance 1 present yet and substance 2 is available, which translates to the Boolean function $b^1(s_1, s_2, s_3) = (1 - s_1) \cdot s_2$. Reaction $r_3$ could be used to model degradation of the substance represented by the first component.*

*The sets of valid reactions are $R_{(0,0,0)} = \{r_4\}$, $R_{(0,0,1)} = \{r_2\}$, $R_{(0,1,0)} = \{r_1, r_4\}$, $R_{(0,1,1)} = \{r_1\}$, $R_{(1,0,0)} = \{r_3\}$, $R_{(1,0,1)} = \{r_2, r_3\}$, $R_{(1,1,0)} = \{r_3\}$, $R_{(1,1,1)} = \{r_3\}$.*

We have not yet defined probability spaces $\Omega_s$, $s \in \mathcal{S}$, for our model. In agreement with Sect. 1.2, we assume that, given a state $s$, an elementary event in $\Omega_s$ basically describes a possible state transition depending on a given transition function. We will give a more specific definition later. In our approach, state transitions depend on the reactions that may be executed in $s$. In general, it is possible for more than one reaction to occur in a given state, and even for a set of reactions to be executed simultaneously. Thus a reaction might be involved in several of the possible state transitions. Mathematically speaking, this amounts to a valid reaction $r = (e, p)$ with $r \in R_s$ describing a probability event $A_r \subseteq \Omega_s$ (*reaction event*) from the probability space $\Omega_s$. That is, reaction events are not necessarily elementary events. Thus, given a state $s$ it is possible that two (or more) reactions $r_1, r_2 \in R_s$ exist with $A_{r_1} \cap A_{r_2} \neq \emptyset$. Therefore, we need to define the sets of reactions which might occur in the same state transition and with this the joint probability for the reaction events.

We call two reactions $r_i$ and $r_j$ *compatible* and denote it with $r_i \sim r_j$ if they satisfy the following two conditions:

$$\begin{aligned} &\exists s \in \mathcal{S} : r_i, r_j \in R_s \,, \\ &\forall s \in \mathcal{S} \text{ with } r_i, r_j \in R_s : (s + e_i + e_j) \in \mathcal{S} \,. \end{aligned}$$

The effect of a set of reactions occurring in a state is calculated as the sum of the single reaction effects. Therefore, we need the compatibility definition of reactions to assure, that this effect does not leave the Boolean state space. This means, reactions which consume or produce the same state variable are not compatible. Note in addition that validity of $r_i$ and $r_j$ in $s$ ensures that corresponding components of the effect vectors have the same value if they are non-zero. Due to this observation, we can easily see that we preserve pairwise compatibility, if we generalize the concept to sets of more than two reactions.

Compatible reactions may occur in the same state transition. We assume that reaction events of compatible reactions in a state $s$ are *stochastically independent*. This is a reasonable assumption from a modeling perspective, since the framework allows to model dependent processes as a single reaction. Furthermore, compatibility ensures that two reactions do not use the same resources or generate the same product. To assign a probability to a reaction event, we define a probability function $P$ on a subset of $2^{\Omega_s}$ for $s \in \mathcal{S}$. The joint probability for the occurrence of two reactions $r_i, r_j \in R_s$ with $r_i = (e_i, p_i)$ and $r_j = (e_j, p_j)$ can then be set as

$$P(A_{r_i} \cap A_{r_j}) = \begin{cases} p_i(s) \cdot p_j(s) & \text{if } r_i \sim r_j \\ 0 & \text{else} \end{cases} .$$

We define the term for more than two compatible reactions accordingly.

If it is essential for modeling purposes to include the possibility of non-compatible reactions being valid in the same state, we have to add one more condition for the choice of reaction probabilities. They need to be defined such that the union of all reaction events in this state is lower or equal one:

$$\forall s \in \mathcal{S} : P(\bigcup_{r \in R_s} A_r) \leq 1.$$

The condition allows for consistent calculation of the probability of unions of reaction events, which is then calculated in the compatible as well as the non-compatible case with the addition formula of probabilities [2]

$$P\Big(\bigcup_{i=1}^{n} A_i\Big) = \sum_{k=1}^{n} (-1)^{k+1} \sum_{\substack{I \subseteq \{1,\ldots,n\}, \\ |I|=k}} P\Big(\bigcap_{i \in I} A_i\Big).$$

**Example 2.3.** *Consider again the running example. It is easy to see that the reactions $r_1$ and $r_4$ are not compatible. However, both reactions might occur in the state $s = (0, 1, 0)$. The union of the two reaction events in the state $s = (0, 1, 0)$ is calculated as:*

$$P(A_1 \cup A_4) = P(A_1) + P(A_4) - P(A_1 \cap A_4) = P(A_1) + P(A_4).$$

*Since the reactions are not compatible, we know that the disjunction of their reaction events is empty. That is, only one or the other reaction can occur and the probabilities for the two reactions must be defined such that $p_1(s) + p_4(s) \leq 1$.*

As already mentioned, reactions are local processes, and sets of reactions might be involved in state transitions. To capture all possible sets of reactions in a given state $s$ we basically have to consider all subsets $M$ of the set of reactions valid in $s$ such that $M$ contains only compatible reactions. This leads to the following definition.

**Definition 2.4.** *Let $\mathcal{R}$ be a set of reactions. A* combination *$C$ is a subset of $\mathcal{R}$ such that there exists $s \in \mathcal{S}$ with $r \in R_s$ for all $r \in C$ and $r_i \sim r_j$ for all $r_i, r_j \in C$ with $i \neq j$.*

It is sufficient to check the compatibility for pairs of reactions in $C$ to ensure that the joint effect of the reactions in $C$ results in a well-defined state. Note furthermore that the empty set $\emptyset$ is always a combination. It has probability greater than zero in a state, if the union of all reaction events in this state is lower than one. This basically amounts to the system remaining in its current state.

We set $C_s := \{C : C \text{ combination}, \forall r \in C\, r \in R_s\}$ the set of valid combinations in a state $s$. Each combination represents a possible state transition. In other word, each combination in a state $s$ represents an elementary event $\omega$ from the probability space $\Omega_s$. This leads to the definition

$$\Omega_s := C_s. \tag{1}$$

We already discussed how to determine the probability of compatible reactions occurring. We can now define a probability function on $\Omega_s$, $s \in \mathcal{S}$, which we call again $P$ in a slight abuse of

| | | | |
|---|---|---|---|
| $s = (0,0,0)$ | $\Omega_s = \{\emptyset, \{r_4\}\}$ | $P(\emptyset) = 1 - p_4(s)$ | $P(\{r_4\}) = p_4(s)$ |
| $s = (0,0,1)$ | $\Omega_s = \{\emptyset, \{r_2\}\}$ | $P(\emptyset) = 1 - p_2(s)$ | $P(\{r_2\}) = p_2(s)$ |
| $s = (0,1,0)$ | $\Omega_s = \{\emptyset, \{r_1\}, \{r_4\}\}$ | $P(\emptyset) = 1 - p_1(s) - p_4(s)$ | $P(\{r_1\}) = p_1(s)$<br>$P(\{r_4\}) = p_4(s)$ |
| $s = (0,1,1)$ | $\Omega_s = \{\emptyset, \{r_1\}\}$ | $P(\emptyset) = 1 - p_1(s)$ | $P(\{r_4\}) = p_1(s)$ |
| $s = (1,0,0)$ | $\Omega_s = \{\emptyset, \{r_3\}\}$ | $P(\emptyset) = 1 - p_3(s)$ | $P(\{r_3\}) = p_3(s)$ |
| $s = (1,0,1)$ | $\Omega_s =$<br>$\{\emptyset, \{r_2\}, \{r_3\}, \{r_2, r_3\}\}$ | $P(\emptyset) = (1 - p_2(s)) \cdot (1 - p_3(s))$<br>$P(\{r_2, r_3\}) = p_2(s) \cdot p_3(s))$ | $P(\{r_2\}) = p_2(s) \cdot (1 - p_3(s))$<br>$P(\{r_3\}) = (1 - p_2(s)) \cdot p_3(s)$ |
| $s = (1,1,0)$ | $\Omega_s = \{\emptyset, \{r_3\}\}$ | $P(\emptyset) = 1 - p_3(s)$ | $P(\{r_3\}) = p_3(s)$ |
| $s = (1,1,1)$ | $\Omega_s = \{\emptyset, \{r_3\}\}$ | $P(\emptyset) = 1 - p_3(s)$ | $P(\{r_3\}) = p_3(s)$ |

Table 1: Probability space definitions for each state of Example 2.1.

notation. The probability of the execution of a combination $C$, that is, the probability of an elementary event in $\Omega_s$, can then be calculated as:

$$P(C) \quad = \quad P\Big(\bigcap_{r \in C} A_r\Big) \cdot \Big(1 - P\Big(\bigcup_{r \in C_s^c} A_r\Big)\Big). \tag{2}$$

Here, the set $C_s^c$ represents the reactions in $R_s$ which are compatible to all reactions in $C$:

$$C_s^c := \{r : r \notin C, r \in R_s, \forall r' \in C : r \sim r'\}. \tag{3}$$

We summarize the key elements of the modeling approach in the following definition.

**Definition 2.5.** *Let $N \in \mathbb{N}$. The pair $\mathfrak{S} = (\mathcal{S}, \mathcal{R})$ with state space $\mathcal{S} := \{0,1\}^N$ and a set of $K \in \mathbb{N}$ reactions $\mathcal{R} := \{r_1, \ldots, r_K\}$ is called a* Stochasticity in Reactions (SIR) *model.*

## 2.2 Dynamics

We represent the dynamics of a SIR model as a probabilistic state transition graph. The transition function $\mathsf{f} : \mathcal{O} \to \mathcal{S}$ calculating the successor of a given state $s$ with respect to a combination $C$ of reactions valid in $s$ is defined as follows:

$$\mathsf{f}(s, C) := s + \sum_{e:(e,p) \in C} e\,.$$

Thus, we consider the graph with vertex set $\mathcal{S}$ and edge set derived from the function

$$\rho : \mathcal{S} \times \mathcal{S} \to [0,1], \quad \rho(s, s') = \sum_{\substack{C \in \Omega_s, \\ \mathsf{f}(s,C) = s'}} P(C)\,,$$

where $P(C)$ is calculated according to Equation 2.

The probabilistic state transition graph for the running example is illustrated in Figure 2. For clarity, the edge labels in the figure do not represent the transition probabilities, but the respective combinations corresponding to a transition. The corresponding probabilities are listed in Table 1.

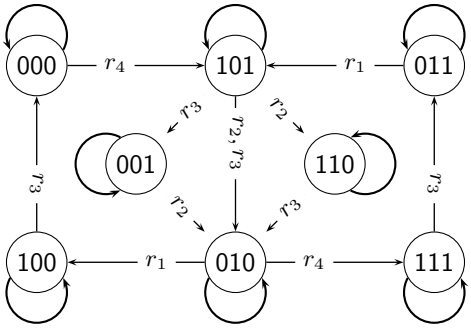



Figure 2: Probabilistic state transition graph for the running example.

A probabilistic state transition graph $G = (V, \rho, E)$ defines a stochastic state transition matrix $\mathcal{P}$ such that $\mathcal{P} := (\rho_{s,s'})$ with $s, s' \in \mathcal{S}$. A *state distribution* $\pi$ is a vector of length $N$ with:

$$\pi \in [0,1]^N \text{ and } \sum_{i=1}^{N} \pi_i = 1\,.$$

An initial state distribution $\pi^\sigma$ assigns each state the probability of this state being the initial state of the system. If we want to start in a specific initial state $s^0 \in \mathcal{S}$, then we choose the distribution $\pi^\sigma$ with $\pi^\sigma_{s^0} = 1$ and $\pi^\sigma_s = 0$ for all $s \neq s^0$.

We are now interested in the sequence of random variables $S(t)$, $t \in \{0, \ldots, T\}$ with $T \in \mathbb{N}$, describing the probability that the system is in state $s$ at time $t \in \{0, \ldots, T\}$. We calculate the probabilities as follows:

$$\begin{aligned} P(S(0) = s) &= \pi^\sigma_s\,, \\ P(S(t+1) = s' \mid S(t) = s) &= \rho_{s,s'}\,. \end{aligned}$$

Such a sequence $S(t)$ is called a *Markov chain* [2]. A realization of the Markov chain is a sequence of states $\boldsymbol{\xi} = (\xi_0, \ldots, \xi_T)$, $\xi_t \in \mathcal{S}$, $t \in \{0, \ldots, T\}$, such that the probability with respect to $\mathcal{P}$ and $\pi^\sigma$

$$P\big((S(0), \ldots, S(T)) = \boldsymbol{\xi} \mid \pi^\sigma\big) = \pi^\sigma_{\xi_0} \cdot \rho_{\xi_1,\xi_2} \cdot \ldots \cdot \rho_{\xi_{T-1},\xi_T}$$

is larger than zero. Such a sequence of states is called trajectory.

### Comparison with other stochastic Boolean frameworks

In the following we show how our formalism relates to the PBN framework described in Sect. 1.2, before we end this section with some general comments. In a PBN model, regardless of the particular state variable update function chosen in a state $s$, the corresponding state variable remains unchanged, increases or decreases in a state transition. For each state variable $n \in N$ we can define an activation reaction $r_n^+ = (e_n^+, p_n^+)$ and a deactivation reaction $r_n^- = (e_n^-, p_n^-)$ where $e_n^+$ is the $n$-th unit vector and $e_n^- = -e_n^+$.

From the PBN definition, activation of a state variable $n$ in a state $s$ may occur, if the state variable is zero and if an update function $f_n^k$ with $k \in 1,\ldots,l(n)$ and $f_n(s) = 1$ exists. A deactivation may occur, if the state variable is zero and if an update function $f_n^k$ with $k \in 1,\ldots,l(n)$ and $f_n(s) = 1$ exists. We can now simply transfer the probabilities in the following way:

$$\begin{aligned} p_n^+ &:= (1 - s_n) \cdot \sum_{\substack{k=1,\\ f_n^k(s)=1}}^{l(n)} c_n^k \,, \\ p_n^- &:= (s_n) \cdot \sum_{\substack{k=1,\\ f_n^k(s)=0}}^{l(n)} c_n^k \,. \end{aligned}$$

This leads to the set of valid reactions in a state $s$

$$\begin{aligned} R_s := \; & \{r_n^+ : n \in \{1,\ldots,N\}, s_n = 0, \exists\, k \in \{1,\ldots,n\} : f_n^k(s) = 1\} \\ & \cup \{r_n^- : n \in \{1,\ldots,N\}, s_n = 1, \exists\, k \in \{1,\ldots,n\} : f_n^k(s) = 0\} \,. \end{aligned}$$

Compatibility is clearly not a problem, thus we can consider the power set of $R_s$, $s \in \mathcal{S}$, as probability space $\Omega_s$. It is easy to see that the PBN and the SIR model generate the same probabilistic state transition graph.

The same reasoning applies when transferring a SIR model into a PBN model, as long as there are no reaction which modify values of two or more state variables. If that is not the case, problems arise since in the PBN formalism the state variable update functions are considered completely independent of each other.

A slight generalization of our formalism allows a much more general observation. Given an arbitrary probabilistic state transition graph, we can construct a SIR model that generates this graph. To obtain this result, we allow for the possibility to artificially declare reactions to be non-compatible. Such a declaration poses no problem from a theoretical point of view, we omitted it for the sake of clarity and due to the page restriction. It leads to smaller sets of valid combinations. Given an arbitrary probabilistic state transition graph, we define a reaction $r$ for each transition from a state $s$ to a state $s'$ in the graph. The effect vector $e$ of $r$ is chosen such that $s+e = s'$, the probability function is chosen such that $r$ is only valid in $s$ and the probability in $s$ matches the transition probability given by the graph. With all reactions being defined to be not compatible, the combinations are just the sets containing only one reaction. Obviously the state transition graph of this model matches the one we started with. In this sense, it is possible to express any model framed in one of the formalisms mentioned in the beginning of Sect. 1.2 as a SIR model, but the idea of reactions as local independent events might get lost.

## 2.3 Analysis

When analyzing a SIR model, we can employ the usual techniques available for Markov chains. Often, we are interested in the state distributions corresponding to certain trajectories. The goal is to identify sequences of state transitions, that is, trajectories in the Boolean state space, that are associated with high probabilities, and thus can be interpreted as most likely behavior of the system. Focussing on trajectories derived from Markov chains, we analyze the corresponding state distributions over time.

Let $\pi(t)$ denote the state distribution of the Markov chain after $t$ time steps, that is, $P(S(t) = s) = \pi(t)_s$. Given an initial state distribution $\pi^\sigma$, it is calculated as:

$$\pi(t) = \pi(t-1)\mathcal{P} \quad \text{with} \quad \pi(0) := \pi^\sigma \,.$$

For more details see e. g. [2]. The probability for the system to be in a certain subset of state space is then calculated as the sum of the corresponding single state probabilities. For example, such a subset might signify an attractor of the system or the set of states where a certain component has value 1.

In addition, existence and reachability of steady states are often points of interest. In the context of Markov chains we consider behavior stable if the state distribution over time converges to a stationary state distribution. If existent the stationary state distribution of a Markov chain is unique. We denote such a stationary state with $\pi^*$. Using the stochastic state transition matrix $\mathcal{P}$, a stationary state satisfies:

$$\pi^* := \pi^*\mathcal{P} \,.$$

Every Markov chain converges to a stationary state distribution, if such a distribution exists. Thus it can be seen as a representation of the long term behavior of a system.

In the remainder of this section we want to focus on analysis of SIR models, where not all or none of the parameters, i. e., the reaction probabilities, are specified. In particular, we are interested in methods to identify parameters. To give an idea of how to approach such a problem we focus on the following very concrete question motivated by the application described in the following section.

Suppose we identified a state $s'$ of particular interest for the functionality of the system. For example, it might be the initial state for some essential process. We now want to choose parameters such that the system favors trajectories reaching the state $s'$. In particular, given a state $s$ we want to maximize the probability of the system to quickly move from $s$ to $s'$. In our formalism, that translates to making sure that certain reactions occur while others are omitted.

To state the problem more precisely, we introduce the following notions. A *direct path* from state $s$ to state $s'$ is a path $\boldsymbol{\tau}_{s,s'} = (\tau_0, \ldots, \tau_l)$, $l \in \mathbb{N}$, in the probabilistic state transition graph such that $\tau_0 = s$, $\tau_l = s'$ and $\tau_i \neq \tau_j$ if $i \neq j$. Note that $\boldsymbol{\tau}_{s,s'}$ then represents a trajectory. Trajectories are called *direct trajectories* if we obtain a direct path by merging consecutive identical states. That is, we allow self-loops in the trajectory, and thus there might exist an infinite number of direct trajectories from $s$ to $s'$ that correspond to the same direct path. We define the probability of a direct path $\boldsymbol{\tau}$ as the sum of the probabilities over all direct trajectories that generate $\boldsymbol{\tau}$ by merging consecutive identical states. We now represent a direct path as the corresponding sequence of non empty combinations. We call such a sequence a *combination sequence*.

**Definition 2.6.** *Given a SIR model $\mathfrak{S} = (\mathcal{S}, \mathcal{R})$, a combination sequence $\mathfrak{r}$ is a finite sequence of non-empty combinations. We say, a combination sequence $\mathfrak{r} := (C^1, \ldots, C^l)$ is valid in a state $s \in \mathcal{S}$ if there exist states $s^1, \ldots, s^l$ with $s = s^1$ such that*

$$C^k \in C_{s^k} \text{ for all } k \in \{1, \ldots, l\} \quad \text{and} \quad s^{k+1} = \mathsf{f}(s^k, C^k) \text{ for all } k \in \{1, \ldots, l-1\}.$$

We now can use combination sequences to calculate the probability of direct paths.

To calculate the probability of a combination sequence, we introduce a random variable $\nu$ representing the combination leading from a state $s$ to a successor $s'$ with $s \neq s'$. In other words, the state variable $\nu$ accepts values in the probability space $\Omega_s^\nu := \Omega_s \backslash \{\emptyset\}$. The probability that a specific combination $C \in C_s \backslash \{\emptyset\}$ is chosen for the next state transition is calculated as:

$$P(\nu = C \mid s) := \frac{P(C)}{P(\bigcup_{C \in C_s \backslash \{\emptyset\}} C)}. \tag{4}$$

**Example 2.7.** *Consider the running example in Figure 2. Given the state* $s = (0, 1, 0)$ *possible combinations in* $s$ *are* $C_s = \{\emptyset, \{r_1\}, \{r_4\}\}$*. The probability that the next state transition is due to the combination* $C = \{r_1\}$ *is:*

$$P(\nu = \{r_1\} \mid (0, 1, 0)) = \frac{P(\{A_1\})}{P(\{A_1\}) + P(\{A_4\})}.$$

Equation (4) allows us to calculate the probability for a specific state transition. The probability of a combination sequence is then calculated as the product of the corresponding transition probabilities.

We now determine all direct paths, and thus combination sequences, from $s$ to $s'$. The sum of their probabilities represents the probability to go from $s$ to $s'$ using a direct path. This leads to a function $F$ which calculates this probability in terms of the (non-specified) reaction probabilities of the included reactions. Maximizing this function leads to a parameter set which maximizes the probability to go from $s$ directly to $s'$, which represents a desirable trait of our system.

When modeling biological systems, we often do not know the exact particulars of all processes involved. The optimization approach introduced above can also be used to compare the impact of different reactions on the system. That is, we basically consider several models that differ in only a small set of reactions and compare the impact of these reactions under parameter constraints derived from optimization for effectiveness of the reactions in question. We illustrate this idea in the following section.

Lastly, we want to mention that it is often reasonable to restrict analysis to suitable subgraphs of the probabilistic state transition graph. Here, we can often exploit the underlying discrete network to identify regions of state space of particular interest.

# 3 The Cytokinin Signal Transduction Network

## 3.1 Biological background

Cytokinin is a plant hormone playing an important role in many developmental and physiological processes in the plant, such as regulation of shoot and root growth, leaf senescence and pathogen resistance. The core of the cytokinin signaling system in *Arabidopsis thaliana* is a multi-step phospho-relay system that influences the expression of a group of target genes. Some of these genes in turn code for regulators of the signaling pathway activity resulting in a negative feedback effect. However, the nature of the feedback mechanism is still unknown. We used the SIR formalism to model the signaling system and test several hypotheses concerning the nature of the feedback mechanism. For a detailed biological background see [5].

The components involved in the signal transduction can be grouped in families of proteins. Proteins from the same family show very similar behavior. To reduce the complexity of the network, we consider the overall behavior of these families, modeling a family as a single network component. In the following we shortly present the different network components.

*Cytokinin*
Cytokinins are a class of plant hormones that influence different important plant functions [8]. Cytokinin initiates the expression of several genes, which are called the cytokinin primary response genes [1].

*Arabidopsis Histidin Kinases (AHKs)*
The AHKs transmit the cytokinin signal from the apoplast into the cell. Cytokinin binds to the ligand binding domain in the extracellular space. This binding causes the canonical histidine residue of the histidine kinase domain to autophosphorylate. After an intramolecular phosphotransfer, the phosphate group can be transferred to an *Arabidopsis* histidin phosphotransfer protein (AHP) via the receiver domain [5, 15].

*Arabidopsis Histidin Phosphotransfer Proteins (AHPs)*
The AHPs act as kinases. AHPs bind to the receiver domain of AHKs and receive their phosphate group. After the phosphorylation of the AHPs, they translocate into the nucleus and transfer their phosphate group to *Arabidopsis* response regulators (ARR) [5, 15]. These can be grouped into two different families.

*Type-B Arabidopsis response regulators (type-B ARRs)*
The type-B ARRs expression seems to be independent of the cytokinin signal. However their activity as transcription factor is directly related to the cytokinin concentration. The phosphorylated type-B ARRs activate the transcription of most cytokinin primary response genes [6].

*Type-A Arabidopsis response regulators (type-A ARRs)*
The type-A ARRs are part of the cytokinin primary response genes. Their transcription is activated by the phosphorylation of the type-B ARR. It has been shown that the type-A ARRs have a (direct or indirect) negative effect on the expression of the cytokinin response genes. Additionally, the phosphorylation of the type-A ARRs results in a higher protein stability.[5, 15]

We focused in our analysis mainly on the character of the feedback mechanism involving type-A ARRs. There exist two theories explaining the negative influence of the type-A ARRs on the cytokinin response. The first theory assumes that the type-A ARRs and the type-B ARRs compete for the phosphate group of AHPs. This is called *AB-competition*. The second theory assumes that active type-A ARRs inhibit the activation of type-B ARRs directly with some unknown mechanism (*AB-inhibition*) [14, 13]. In the following, both theories will be considered.

## 3.2 SIR Model

### State space

We model the components corresponding to cytokinin, AHK, AHP and type-B ARR as Boolean variables. Cytokinin is either *absent* or *present* and the three components AHK, AHP and type-B ARR are either *inactive* or *active*. We need to be more precise in the case of type-A ARRs. Type-A ARRs are either *absent*, *present/inactive*, or *present/active*. This is modeled using two

| | | |
|---|---|---|
| $\mathcal{S}_1$ | : | Cytokinin ($C$) |
| $\mathcal{S}_2$ | : | AHK ($K$) |
| $\mathcal{S}_3$ | : | AHP ($P$) |
| $\mathcal{S}_4$ | : | Type-B ($B$) |
| $\mathcal{S}_5$ | : | Type-A ARR inactive ($A$) |
| $\mathcal{S}_6$ | : | Type-A ARR active ($Aa$) |

Table 2: Components of the cytokinin signaling network.

Boolean variables. The first state variable indicates presence or absence of type-A ARRs and the second specifies their activity status. Type-A ARRs will not be present in their active and their inactive form at the same time. The network component are listed in Table 2 with the associated variable name in parenthesis.

**Reactions**

Since we want to consider different manifestations of the negative feedback mechanism involving type-A ARRs, we have to consider different definitions for the corresponding reactions. However, several reactions are independent of the assumptions concerning type-ARRs negative feedback. For illustrative purposes, we explain the modeling of some reactions in detail. All specifications are listed in Table 3.

*Activation of AHK:* $K^{(+)}$
AHK autophosphorylates in the presence of cytokinin. This reaction requires the inactive form of the AHK. The two conditions are encoded in the Boolean function $b^{K^{(+)}}(s) = s_1 \cdot (1 - s_2)$ partly defining the probability function. The reaction results in an activation of the AHK. This reaction effects the value of the state variable $K$ only. This is reflected in its effect vector $e_{K^{(+)}} = (0, 1, 0, 0, 0, 0)$.

*Activation of AHP:* $P^{(+)}$
AHP becomes phosphorylated if AHK is phosphorylated. This reaction requires AHK to be in the active and AHP to be in the inactive form as reflected by the Boolean function $b^{P^{(+)}}(s) = s_2 \cdot (1 - s_3)$. The reaction results in an inactivation of the AHK and an activation of the AHP. This reaction increases the value of state variable $P$ and decreases the value from $K$ as can be seen in its effect vector $(0, -1, 1, 0, 0, 0)$.

*Deactivation of type-B ARR:* $B^{(-)}$
Type-B ARR tend to be unstable in their phosphorylated form. An explanation for this effect is a dephosphorylation reaction involving no other network components. The resulting effect vector is $(0, 0, 0, -1, 0, 0)$. The reaction requires type-B ARR to be in their active form, thus $b^{B^{(-)}}(s) = s_4$.

*Expression of type-A ARR:* $A_e$
The expression of type-A ARR is induced by the phosphorylated type-B ARR. This reaction requires type-B ARR to be in their active form and type-A ARR to be absent.

*Degradation of inactive type-A ARR:* $A_d$
Inactive type-A ARR might degrade. This reaction requires inactive type-A ARR.

*Activation of type-A ARR:* $A^{(+)}$
Type-A ARR become phosphorylated as AHP is phosphorylated. This reaction requires AHP to be in their active form and type-A ARR to be in their inactive form. It causes an activation of type-A ARR.

*Degradation of active type-A ARR:* $Aa_d$
Degradation of active type-A ARR might proceed at a different rate than degradation of inactive type-A ARR. Therefore, it is modeled as a distinct reaction. This reaction requires active type-A ARR.

**Modeling feedback mechanisms**

*Activation of type-B ARR:*$B^{(+)}$
The two type-A ARR negative feedback assumptions effect type-B ARR activation. Therefore, we consider two different reaction definitions for the two assumptions. As a control we also consider a reaction definition without negative feedback.

*Version one: no negative feedback*: $B^{(+)1}$
Type-B ARR become phosphorylated as AHP are active. This reaction requires the active form of AHP and an inactive form of type-B ARR. It results in an inactivation of AHP and an activation of type-B ARR.

*Version two: competition*: $B^{(+)2}$
Type-A ARR and type-B ARR compete for AHP. Due to this consideration, phosphorylation of the type-B ARR is prevented in the presence of inactive type-A ARR.

*Version three: inhibition*: $B^{(+)3}$
Type-A ARR inhibits phosphorylation of type-B ARR. Phosphorylation of type-B ARR is prevented in the presence of active type-A ARR.

Due to the different considerations about the type-A ARR negative effect on the cytokinin response, we consider three different reaction sets, which in turn signify three different models:

$$\begin{aligned}\mathcal{R}_1 &= \{K^{(+)}, P^{(+)}, B^{(+)1}, B^{(-)}, A_e, A_d, A^{(+)}, Aa_d\}\,,\\ \mathcal{R}_2 &= \{K^{(+)}, P^{(+)}, B^{(+)2}, B^{(-)}, A_e, A_d, A^{(+)}, Aa_d\}\,,\\ \mathcal{R}_3 &= \{K^{(+)}, P^{(+)}, B^{(+)3}, B^{(-)}, A_e, A_d, A^{(+)}, Aa_d\}\,.\end{aligned}$$

We do not yet specify the reaction probabilities. Rather, considerations concerning the stochastic parameters are integrated in the system analysis presented in the following.

## 3.3 Analysis and Results

We want to compare the different assumptions of competition and inhibition in this part. Therefore, we determine the states in which the two effects influence the type-B ARR activation and analyze the influence of the reaction probabilities for reaching these states. We choose the state $s^0 = (1, 0, 0, 0, 0, 0)$ as initial state, representing a cytokinin signal and a quiescent state for all components of the signaling pathway. It is easy to see that the first transitions are fully determined by this state. That is, the first reaction that occurs is reaction $K^{(+)}$ followed by $P^{(+)}$. Stochastic aspects come into play following the execution of these reactions.

$$
\begin{array}{lcl}
p_{K^{(+)}}(s) & = & pr_{K^{(+)}} \cdot (s_1 \cdot (1 - s_2)) \\
e_{K^{(+)}} & = & (0, 1, 0, 0, 0, 0) \\
p_{P^{(+)}}(s) & = & pr_{P^{(+)}} \cdot (s_2 \cdot (1 - s_3)) \\
e_{P^{(+)}} & = & (0, -1, 1, 0, 0, 0) \\
p_{B^{(+)1}}(s) & = & pr_{B^{(+)1}} \cdot (s_3 \cdot (1 - s_4)) \\
e_{B^{(+)1}} & = & (0, 0, -1, 1, 0, 0) \\
p_{B^{(+)2}}(s) & = & pr_{B^{(+)2}} \cdot (s_3 \cdot (1 - s_4) \cdot (1 - s_5)) \\
e_{B^{(+)2}} & = & (0, 0, -1, 1, 0, 0) \\
p_{B^{(+)3}}(s) & = & pr_{B^{(+)3}} \cdot (s_3 \cdot (1 - s_4) \cdot (1 - s_6)) \\
e_{B^{(+)3}} & = & (0, 0, -1, 1, 0, 0) \\
p_{B^{(-)}}(s) & = & pr_{B^{(-)}} \cdot (s_4) \\
e_{B^{(-)}} & = & (0, 0, 0, -1, 0, 0) \\
p_{A_e}(s) & = & pr_{A_e} \cdot (s_4 \cdot (1 - s_5) \cdot (1 - s_6)) \\
e_{A_e} & = & (0, 0, 0, 0, 1, 0) \\
p_{A_d}(s) & = & pr_{Aa_d} \cdot (s_6) \\
e_{A_d} & = & (0, 0, 0, 0, 0, -1) \\
p_{A^{(+)}}(s) & = & pr_{A^{(+)}} \cdot (s_3 \cdot s_5) \\
e_{A^{(+)}} & = & (0, 0, -1, 0, 0, 1) \\
p_{Aa_d}(s) & = & pr_{Aa_d} \cdot (s_6) \\
e_{Aa_d} & = & (0, 0, 0, 0, 0, -1)
\end{array}
$$

Table 3: Reaction definitions for the cytokinin SIR model.

We proceed as follows. First, we focus on the model representing competition. We consider the set of states for which the reaction modeling the competition is valid, determine the impact of competition on state transitions starting in such states and analyze the influence of reaction probabilities on the probability that the system reaches such states. Maximizing the probability for reaching states where competition may come into play yield optimal parameters for effective competition. We apply the methods utilizing direct paths, which we introduced in Sect. 2.3.

Competition effects the behavior of the system if the system is in a state with type-A ARR and type-B ARR inactive. If cytokinin is present, as we assume throughout this analysis, there is a total number of four such states. All of them are reached from states with active type-B ARR and absent type-A ARR by executing the expression reaction of type-A ARR and deactivation of type-B ARR. Furthermore, the expression of type-A need to occur before or at the same time as the type-B inactivation. Type-B ARR deactivation should occur before type-A activation and before type-A degradation. We assume that $K^{(+)}$ and $P^{(+)}$ are very fast reactions, which is in agreement with biological observations. So whenever $K^{(+)}$ or $P^{(+)}$ are valid reactions they should be immediately executed. This can be modeled by assigning them probability 1. This assumption allows us to determine the influence of the reactions $A_e$, $A_d$ $A^{(+)}$ and $B^{(-)}$ on the accessibility of the competition state by focussing on a subgraph of the state transition graph.

Figure 3 shows the resulting subgraph of the state transition graph, which is induced by the three states $s^1 = (1, 1, 1, 1, 0, 0)$, $s^2 = (1, 1, 1, 1, 1, 0)$ and $s^3 = (1, 1, 1, 0, 1, 0)$. For clarity, we represent each state only by the components corresponding to variables $B$, $A$ and $Aa$. The state $(0, 1, 0)$ represents the state of interest for competition. Edges represent the possible reaction combinations in the states. Edges without head vertex represent combinations which are valid

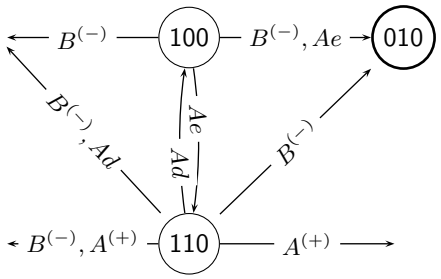



Figure 3: A part of the model state transition graph, for the state variables $B$, $A$ and $Aa$ with $C = 1$, $K = 1$ and $P = 1$

in the states but do not result in a state of this subgraph. Starting with active type-B ARR and absent type-A ARR, we now want to calculate the probability of the system evolving from $(1, 0, 0)$ to $(0, 1, 0)$ without leaving the subgraph. There are two reaction sequences that represent this behavior $\mathfrak{r}_1 = (\{A_e, B^{(-)}\})$ and $\mathfrak{r}_2 = (\{A_e\}, \{B^{(-)}\})$:

$$(1,0,0) \xrightarrow{A_e, B^{(-)}} (0,1,0) \qquad \text{and} \qquad (1,0,0) \xrightarrow{A_e} (1,1,0) \xrightarrow{B^{(-)}} (0,1,0).$$

The probability of the reaction sequence $\mathfrak{r}_1$ is calculated using the function $\mathfrak{f}^1 : [0,1]^4 \to [0,1]$ with:

$$\mathfrak{f}^1(pr_{A_e}, pr_{A_d}, pr_{A^{(+)}}, pr_{B^{(-)}}) = \frac{pr_{B^{(-)}} \cdot pr_{Ae}}{pr_{B^{(-)}} + pr_{Ae} - pr_{B^{(-)}} \cdot pr_{Ae}}.$$

The probability of the reaction sequence $\mathfrak{r}_2$ is calculated using the function $\mathfrak{f}^2 : [0,1]^4 \to [0,1]$ with:

$$\begin{aligned} &\mathfrak{f}^2(pr_{A_e}, pr_{A_d}, pr_{A^{(+)}}, pr_{B^{(-)}}) \\ = \;& \frac{pr_{Ae} \cdot (1 - pr_{B^{(-)}})}{pr_{Ae} + pr_{B^{(-)}} - pr_{B^{(-)}} \cdot pr_{Ae}} \times \frac{pr_{B^{(-)}}(1 - pr_{Ad} - pr_{A^{(-)}})}{pr_{B^{(-)}} + pr_{Ad} + pr_{A^{(-)}} - pr_{B^{(-)}} \cdot (pr_{Ad} + pr_{A^{(-)}})}. \end{aligned}$$

We now proceed with an optimization approach as described in Sect. 2.3 to identify parameters that favor competition. This allows us to analyze the maximal effect of the competition model in terms of negative feedback. To determine optimal conditions for the competition, we optimize the function $F$ with:

$$F(pr_{A_e}, pr_{A_d}, pr_{A^{(+)}}, pr_{B^{(-)}}) = f^1(pr_{A_e}, pr_{A_d}, pr_{A^{(+)}}, pr_{B^{(-)}}) + f^2(pr_{A_e}, pr_{A_d}, pr_{A^{(+)}} pr_{B^{(-)}}),$$

$$\text{while } pr_{A_e}, pr_{A_d}, pr_{A^{(+)}}, pr_{B^{(-)}} \in [0.01, 0.5]\,.$$

Here, we restrict the probability parameter to the interval $[0.01, 0.5]$ in order to allow a minimal reaction probability of 0.01 and to allow a minimal stochastic effect of 0.5. We calculate the maximum of $F$ using a numerical solution algorithm, as implemented for example in MATLAB. Optimization yields a vector $\theta = (0.01, 0.5, 0.01, 0.1429)$ of probability values with $F(\theta) = 0.78$.

Clearly, the calculated probabilities for the reactions $A_d$ and $A^{(+)}$ are at the minimal value of 0.01 while the probability for the reaction $A_e$ is at the maximal possible value with 0.5. This

reflects that for effective competition reactions $A_d$ and $A^{(+)}$ should not be executed in the states considered, while reaction $A_e$ should occur rapidly. The probability of $B^{(-)}$ favors the outcome of $B^{(-)}$ occurring before reactions $A_d$ and $A^{(+)}$, but after $A_e$.

In the following we analyze the behavior of the three models using the parameters favoring competition. A direct comparison is then possible since we use the same parameter set on all three cases. Since we chose the parameters to optimize competition, we would expect that the corresponding model displays strong negative feedback effects, while the other models might show a weaker or no effect at all.

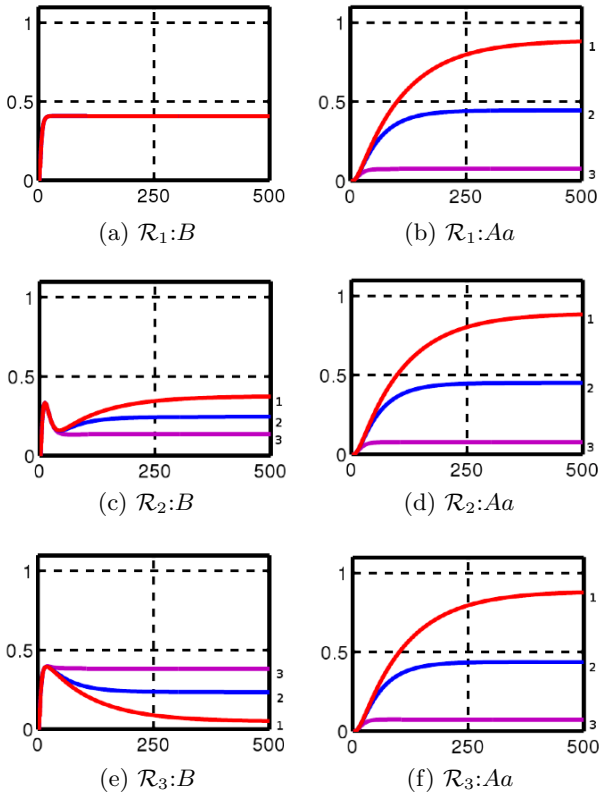
The inhibition mechanism prevents type-B ARR phosphorylation if type-A ARR is active until degradation of the type-A ARR. Therefore the effectiveness of the inhibition is highly dependent on the probability of type-A ARR activation and the probability of the active type-A ARR degradation. We consider three different values for active type-A ARR degradation for the analysis. The parameter specifications are summarized in the following table.

| | | | |
|---|---|---|---|
| $pr_{K^{(+)}} =$ | 1 | $pr_{A_e} =$ | 0.5 |
| $pr_{P^{(+)}} =$ | 1 | $pr_{A^{(+)}} =$ | 0.01 |
| $pr_{B^{(+)}} =$ | 0.1 | $pr_{A_d} =$ | 0.01 |
| $pr_{B^{(-)}} =$ | 0.1429 | $pr_{Aa_d} \in$ | $\{0.001, 0.01, 0.1\}$ |

For our analysis we focus on the behavior of the components $B$ and $Aa$ since they are directly involved in the different mechanisms. Figure 4 shows state variable activity for the state variables $B$ and $Aa$ for the reaction sets $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$. Here, state variable activity denotes the probability of the system to be in a state where the corresponding state variable has value one. Thus, the curves in the figure illustrate the probability of occurrence of active type-B ARR and active type-A ARR. We used the initial state distribution $\pi^\sigma$ specifying $s^0 = (1, 0, 0, 0, 0, 0)$ as initial state, i. e., $\pi^\sigma_{s^0} = 1$, and plotted the probabilities over 500 time steps. The different curves in one coordinate system correspond to different choices for the probability of active type-A degradation.

The figure shows that $Aa$ activity hardly varies depending on the choice of model, while a strong impact of the choice of probability value for its degradation can be observed. Activity of $B$ is completely independent of $Aa$ activity for reaction set $\mathcal{R}_1$ due to the absence of negative influence in the control model. For reaction set $\mathcal{R}_2$, modeling competition, higher stability of active type-A ARR (modeled by lower probabilities for $Aa$ degradation) influences $B$ activity in the beginning, yet ongoing observation reveals a stronger inhibition of $B$ activity if we assume low stability of active type-A ARR. Lastly, in the inhibition model effectiveness of the inhibition increases with increasing stability of $Aa$.

The results show that competition shows a strong inhibitory effect on type-B ARR activity only if stability of type-A ARR is rather low. In contrast, the inhibitory mechanism is the more effective the higher the stability of $Aa$. Based on these observations, further biological experiments relating stability of active type-A ARR to regulation of the cytokinin response genes might help to clarify the nature of the negative feedback mechanism. Earlier experimental results indicate that type-A ARR stability increases with activation [13]. Together with our calculation, this allows for a tentative hypothesis favoring the inhibitory mechanism.




(a) $\mathcal{R}_1$:$B$ (b) $\mathcal{R}_1$:$Aa$

(c) $\mathcal{R}_2$:$B$ (d) $\mathcal{R}_2$:$Aa$

(e) $\mathcal{R}_3$:$B$ (f) $\mathcal{R}_3$:$Aa$

Figure 4: State variable activity of $B$ and $Aa$ for the three reaction sets $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$ and with three different active type-A degradation probabilities: $1 : pr_{Aa_d} = 0.001$, $2 : pr_{Aa_d} = 0.01$ and $3 : pr_{Aa_d} = 0.1$

## 4 Discussion

In this paper we present a new method to model complex interactions in biological networks using a hybrid framework combining Boolean modeling and stochastic effects. Compared to other probabilistic Boolean models, our method is based on a more local approach, modeling reactions usually only involving a small subset of the system's components. This allows for a very flexible choice with regard to the update strategy determining state transitions in the Boolean state space. Reactions group together changes in component values that are dependent on each other and thus should be executed at the same time. It is furthermore possible to group reactions together in order to consider simultaneous effects of different reactions as well.

Probabilities assigned to reactions allow for modeling of several important aspects influencing the behavior of a biological system that cannot be captured by a purely Boolean approach. Reaction probabilities may represent uncertainties in the execution of processes due to environmental conditions or faulty realization, but they can also be used to distinguish fast and slow processes. Analysis of the resulting probabilistic state transition graph can focus on a variety of aspects, for example determination of trajectories with high probabilities or examination of the importance of a given reaction for the system's dynamics. Since parameter identification is clearly an issue, analysis of unspecified models can be carried out in order to obtain statements

relating parameter values to possible behaviors. We plan to investigate possible approaches to such questions in future work.

We used the Markov property to simulate the dynamic of the system, which allows for effective simulation and analysis exploiting the rich theory and existing tools for Markov chains. However, the Markov property is a very strong assumption in the context of modeling biological systems, since processes of different time scales might become effective when taking into account accumulation effects. This difficulty can be addressed using additional state variables simulating specific memory effects. However, the addition of state variables should be limited due to the exponential growth of the state space and the probabilistic state transition graph. To balance the effects, methods focussing on analysis of submatrices of the state transition matrix, representing independent modules of the system, need to be studied in more detail. It might also be fruitful to study the properties of the underlying discrete model more closely to exploit available network reduction methods.

The presented formalism already proved useful in application. The mechanism by which the negative feedback loop of cytokinin signaling works has been much discussed but remains obscure. The work presented here marks the protein stability of the type-A ARRs as a decisive factor determining whether inhibition or competition is more likely. Protein stability of the type-A ARRs has not been previously considered in this context. Thus the results of this study provide us with a new experimentally testable hypothesis.

## References

[1] W. G. Brenner, G. A. Romanov, I. Köllmer, L. Bürkle, and T. Schmülling. Immediate-early and delayed cytokinin response genes of arabidopsis thaliana identified by genome-wide expression profiling reveal novel cytokinin-sensitive processes and suggest cytokinin action through transcriptional cascades. *Plant J.*, 44(2):314–333, 2005.

[2] K. L. Chung. *Elementary Probability Theory with Stochastic Processes*. Springer-Verlag, 1974.

[3] H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselmann. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull. Math. Biol.*, 66(2):301–340, 2004.

[4] A. Garg, K. Mohanram, A. D. Cara, G. D. Micheli, and I. Xenarios. Modeling stochasticity and robustness in gene regulatory networks. *Bioinformatics*, 25(12):i101–i109, 2009.

[5] A. Heyl and T. Schmülling. Cytokinin signal perception and transduction. *Curr. Opin. Plant Biol.*, 6(5):480–488, 2003.

[6] I. Hwang and J. Sheen. Two-component circuitry in arabidopsis cytokinin signal transduction. *Nature*, 413(6854):383–389, 2001.

[7] S. A. Kauffman. *Origins of Order*. Oxford University Press, 1993.

[8] D. W. Mok and M. C. Mok. Cytokinin metabolism and action. *Annu. Rev. Plant Biol.*, 52:89–118, 2001.

[9] A. Ribeiro, R. Zhu, and S. A. Kauffman. A general modeling strategy for gene regulatory networks with stochastic dynamics. *J. Comput. Biol.*, 13(9):1630–1639, 2006.

[10] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, 2002.

[11] H. Siebert and A. Bockmayr. Temporal constraints in the logical analysis of regulatory networks. *Theoret. Comput. Sci.*, 391:258–275, 2008.

[12] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. ii. logical analysis of regulatory networks in terms of feedback circuits. *Chaos*, 11(1):180–195, 2001.

[13] J. P. C. To, J. Deruère, B. B. Maxwell, V. F. Morris, C. E. Hutchison, F. J. Ferreira, G. E. Schaller, and J. J. Kieber. Cytokinin regulates type-a arabidopsis response regulator activity and protein stability via two-component phosphorelay. *Plant Cell*, 19(12):3901–3914, 2007.

[14] J. P. C. To, G. Haberer, F. J. Ferreira, J. Deruère, M. G. Mason, G. E. Schaller, J. M. Alonso, J. R. Ecker, and J. J. Kieber. Type-a arabidopsis response regulators are partially redundant negative regulators of cytokinin signaling. *Plant Cell*, 16(3):658–671, 2004.

[15] J. P. C. To and J. J. Kieber. Cytokinin signaling: two-components and more. *Trends Plant Sci.*, 13(2):85–92, 2008.

[16] L. Tournier and M. Chaves. Uncovering operational interactions in genetic networks using asynchronous boolean dynamics. *J. Theoret. Biol.*, 260(2):196–209, 2009.