

Discrete, qualitative models of interaction networks

Kathrin Ballerstein¹, Utz-Uwe Haus¹, Jonathan Axel Lindquist¹, Tilo Beyer², Burkhardt Schraven², Robert Weismantel¹

¹*Institute of Operations Research, Department of Mathematics, ETH Zurich, Raemistrasse 101, CH-8092 Zurich, Switzerland,*

²*Institute for Molecular and Clinical Immunology, Medical Faculty, Otto-von-Guericke-Universitaet, Leipziger Str. 44, Building 26, 39120 Magdeburg, Germany*

TABLE OF CONTENTS

1. Abstract
2. Introduction
 - 2.1. Propagation techniques
 - 2.2. Interaction graphs
 - 2.3. Kinetic logic and Petri nets
3. Logic framework for interaction networks
 - 3.1. From blots to formulas
 - 3.2. Structural analysis
 - 3.3. Functional analysis
4. Dynamic model
5. Multiple activation levels
6. Conclusions
7. References

1. ABSTRACT

Logical models for cellular signaling networks are recently attracting wide interest: Their ability to integrate qualitative information at different biological levels, from receptor-ligand interactions to gene-regulatory networks, is becoming essential for understanding complex signaling behavior. We present an overview of Boolean modeling paradigms and discuss in detail an approach based on causal logical interactions that yields descriptive and predictive signaling network models. Our approach offers a mathematically well-defined concept, improving the efficiency of analytical tools to meet the demand of large-scale data sets, and can be extended into various directions to include timing information as well as multiple discrete values for components.

2. INTRODUCTION

The development of new technologies and high throughput methods provides new insights in the structure and function of biological units. In response to this growing complexity, the need for structural and functional theoretical analysis has emerged. Relevant questions such as predicting a cell's response, determining suitable intervention strategies for dysfunctions, or simply detecting modeling errors require answers beyond trial-and-error.

When speaking about biological systems we refer to any level of detail. In particular, it can mean a mechanistic description of a cell stating all (known) interactions on a molecular basis, or it can denote a schematic model of the development of a disease. The

Discrete, qualitative models of interaction networks

interpretation strongly depends on the purpose of the model and the quality of data. In accordance, we always name an element in the biological system component and for a reaction or biological consequence we use the term interaction. The termini active and inhibited refer to characteristics such as the right or wrong location for an interaction, enzymatic activity, phosphorylated or dephosphorylated forms, or the sufficient presence or absence of a protein. We will not further discuss these details, other than to state that for each individual model they need to be carefully specified for every component.

Several different modeling techniques were introduced to analyze biological systems. Many approaches utilize ordinary differential equations to describe small, quantitative systems with respect to reaction rates and production levels (for an overview see (1)). One famous concept which is still widely used for quantitative analysis is Michaelis-Menten kinetics (2). Due to the high complexity to solve differential equation systems this approach is not appropriate for every system. Therefore, it is often applied only to well-characterized subsystems. However, for many biological systems there is not sufficient data available to allow a quantitative approach. Often one can only distinguish between two states, e.g. expressed or not expressed, which does not permit a reasonable parameter estimation.

Here, we will concentrate on qualitative concepts to model a biological units functionality. We will start with an overview of established qualitative modeling methods, then provide a self-contained presentation of our modeling formalism based on logical implications and discuss its extension to a dynamic setting: In Section 3.1 we show how to systematically derive implication formulas from measurements, then discuss structural analysis of the underlying interaction network (Sec. 3.2), and show how to make functional predictions (Sec. 3.3 and 4). Therefore, Sections 3 and 4 can be read as a guideline for users that first leads the way to a logical model of the biological system and secondly explains the potentials of the model including solving techniques and their complexity. Finally, we will introduce a new approach to the situation where multiple activation levels are to be modeled without falling back to a reformulation with binary variables.

2.1. Propagation techniques

A first step towards qualitative modeling techniques was accomplished by Kauffman in 1969 (3). He represented each component in the biological unit by a binary variable whose value is determined by a propositional formula. Given an initial set of values for all variables, the evolution of the system's states is computed by evaluating the logical formulas for one state to obtain the next. Descriptively speaking the values are propagated through the system just like reactions occurring one after the other. In further work Glass and Kauffman (4) introduce *steady states* of a logical system. They form an analog of the steady state in the continuous setting, which is defined as a state where the first derivative of all concentration functions is 0. In the discrete case steady states are defined as a state that has itself as an ensuing

state, i.e. evaluating the propositional formulas for the current setting of 0 and 1 yields the identical pattern. It has been shown that following a certain mapping between discrete and continuous models the two concepts coincide. These considerations form the basis for the concept of *Boolean networks* which is a widely used model. Many extensions and applicable analytic tools have so far been proposed (e.g. (5,6,7)).

2.2. Interaction graphs

Analyzing a biological system by means of Boolean networks has two dimensions. The first is a structural analysis which utilizes a graphical representation of the system, the so called *interaction graph* (8). It is not only used to visualize the system, but can also answer questions concerning connectivity, which component influences which, and identify feedback loops, i.e. subsystems where a component influences itself positively or negatively via several other components. Applying graph theoretical and combinatorial optimization tools these questions can directly be investigated (9,8). These tools are limited in the sense that they ignore the type of interaction, which becomes critical when multiple components are required to activate another one, a so called AND connection. To overcome this restriction Klamt *et al.* (8) generalize their concept of interaction graphs to directed hypergraphs. This concept allows not only arrows between two components, but they are forked so that one arc has several tails leading into one component. Hence, a directed hyperarc represents an AND connection. However, efficient graph theoretic methods to analyze the system directed hypergraphs are rare.

The second aspect of describing biological systems by Boolean models concerns the functionality. To determine steady states an initial state is assumed, for which all interactions are evaluated, yielding the next state (8). This iterative process is continued until a steady state is reached. Therefore a steady state is an assignment of values to each component that does not change when all interactions are evaluated again, just as in Kauffman's concept (3). A particular application are minimal intervention sets in which the value of some components are predetermined. Then all compatible steady states are computed as described above and analyzed with respect to components that do not change their value. Thus, minimal intervention sets are a key instrument for *in silico* drug development. The computation involves listing and testing all possible sets, which implies a limited size of computable intervention sets. In general, the analysis of Boolean networks can be applied to large-scale systems, and is therefore suitable for systems with many components where few details about the interactions are known.

2.3. Kinetic logic and Petri nets

Another modeling paradigm is that of *Petri Nets*, which will not be presented here in detail. It is a separate field of research containing many extensions and variations of the classical concept of Petri Nets. Often it is applied to small and medium sized systems. More information can be found in the PhD. thesis of Petri (11), a recent overview with applications to systems biology is given in (12).

Discrete, qualitative models of interaction networks

Table 1. Overview of the basic terms of propositional logic

Symbol	Logical meaning	Interpretation
component		
\wedge	conjunction	logical AND
\vee	disjunction	logical OR
\neg	negation	logical NOT
$A \rightarrow B$	implicatio	
$A \leftrightarrow B$	equivalence	$(\neg A \vee B) \wedge (A \vee \neg B)$
$a \approx b$	logical equivalence	two logical terms can be transformed into each other

A closely related strategy by Thomas (13) to handle discrete data formalizes a so called *kinetic logic* for gene regulatory networks. It is based on the theory of Boolean control circuits and the difference compared to Petri Nets is to involve reaction times and delays, which makes it a qualitative concept closely related to the differential equations techniques. It deals with interactions in much more detail. Two types of logical variables are distinguished: *environmental* variables, that model the environmental influences like temperature or mutation type, and *internal* variables, which serve as a memory variable for the current level of the respective element. For example, this can be the concentration of a product. For each internal variab x he associates an internal Boolean *function* X describing the regulation of the corresponding gene \bar{x} . As a first step the internal functions are manually determined. Thus, a model for the gene regulatory network is established. As demonstrated in (14) the system can then be simulated with all possible values for the environmental and internal variables in order to obtain predictions and explanations for the systems behavior. Identifying steady states or oscillating states is the main interest rather than structural properties as this approach is best suited for small networks. A system can be in two different states, stable or unstable/transient: If the given values of the internal variables coincide with the evaluated internal function for specific, fixed values of the environmental variables, then it is in a stable state. Otherwise, it is in a transient state. Biologically, the idea behind this is to include the duration of biological processes. If a gene \bar{g} is present above a certain threshold, so that it is able to perform its function, it will still take some time until its product \bar{p} is actually present in functionally relevant amounts. In this case, $g = 1$ and the associated function $P = g$ delivers a value of 1. So the system is in a transient state, as $g = 1$ but still $p = 0$. After a certain time, p will shift over to 1 which brings the system back to a stable state. The evolution of the biological system over time for a given environmental influence can then be illustrated by a state transition graph. Before we describe the approach by an example, we review the basic terms of propositional logic in Table 1. For the reader unfamiliar with the formalism of propositional logic we also refer to (15).

Example 1 (taken from (14)). Suppose we are given a biological system described by two internal functions

$$X = \neg y \vee \neg o$$

$$Y = \neg x \vee t$$

where x, y are internal variables and o, t are environmental variables, say oxygen supply and temperature. Function X states that the corresponding gene \bar{x} is active if y is absent or the repressor, which corresponds to the site where gene y represses gene x , is inactive ($o = 0$). Accordingly, the gene \bar{y} associated with Y is only active if x is absent or the temperature is high ($t = 1$). Note, that o and t are determined by the user depending on the experiment. Following the approach of Thomas the system is simulated with all possible truth values of x, y, o , and t which can be found in Table 2. In the highlighted scenarios the actual value of X and Y equal those of x and y and therefore the system is stable. Consider, e.g. $(x, y, o, t) = (0, 1, 0, 0)$, since $o = 0$, X evaluates to 1 and the gene \bar{x} can be switched on. But the reaction takes some time to produce \bar{x} , and hence x will only be 1 at a later time point, bringing us to the state $(x, y, o, t) = (1, 1, 0, 0)$. This is again transient. With an analogous argument y shifts to 0, finally yielding a stable state. Note that in the last column we find two stable states. Which one of them is achieved depends on the duration of the reactions and the starting point. In Figure 1 the state transition graph for this situation $(o, t) = (1, 0)$ illustrates the competition of the two stable states fought by the production/degradation times $t_x^{pro}, t_y^{pro}, t_x^{deg}$, and t_y^{deg} .

A detailed exposition of the kinetic logic concept and its applications can be found in (14). Some refinements of this approach are known (e. g. (16, 17, 18, 19)). On the one hand discrete quantification of the variables are included by e.g. Van Ham (20), who encodes different levels of the variables as binary variables for each level. Such incidence variables take the value 1 if the gene exceeds the associated level and 0 otherwise. On the other hand more detailed time information are incorporated. Siebert and Bockmayr (21) for example allow for specific durations of each reaction using the framework of timed automata. However, this approach and its extensions are limited by the size of a system: One has to compute the values of all internal functions for every possible scenario of internal and environmental variables and compare them with each other.

In general, the question which model is best suited for a biological system depends on the available data and the type of question studied: More detailed models on the one hand elucidate the functionality of the system to a higher extent, but on the other hand reduced, qualitative models may reveal macroscopic behavior just as well with much smaller measuring, modeling, and computational effort.

Table 2. Simulation of the Thomas system for all possible values of x, y, O , and t .

XY	00	01	11	10	O, t
00	11	11	11	11	
01	11	11	01	01	
11	10	11	01	00	
10	10	11	11	10	
x, y					

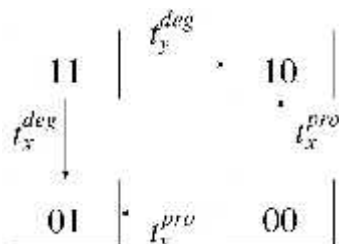


Figure 1. The state transition graph for $(o,t)=(1,0)$ in Example 1 where t^{pro} denotes the production and t^{deg} the degradation time of the associated gene.

3. LOGIC FRAMEWORK FOR INTERACTION NETWORKS

Over the last years large-scale systems have gained considerable importance to users as descriptive and predictive models to improve insight into structure and functionality (e. g. (22, 23, 24, 25, 26) or see the reviews (27, 28)). The logic framework presented in this section is related to the concept of Boolean networks. It aims at strictly formalizing the concept, improving the efficiency of analytical tools to meet the demand of large-scale data sets, and extending the framework into various directions, including temporal information as well as multiple discrete values for each component. We will consider interaction networks as bipartite graphs, introduce the qualitative, logic framework, and expand this by including temporal information and a discretization of activation levels. Theoretical results concerning complexity and structure of the developed model are displayed and are illustrated by means of practical applications.

3.1. From blots to formulas

The simplest model of a biological process is that which one obtains by collecting diverse knowledge about the various individual causal relationships of its constituent components. This is in fact the common implicit modeling paradigm employed by users that argue about a biological process using their accumulated knowledge about the relevant entities. We will follow this path in a well-formalized way: The intuitive model of a biological process is to describe all (causal) experimental statements in the form of logical formulas, that can be written down in propositional logic (29): Introduce component variables for each component, and write down implication formulas for experimentally proven knowledge statements like “*MEK activates ERK*” as

$$MEK \rightarrow ERK$$

and “*In the absence of (activated) PTEN and SHIP1 we find that PI3K generates an increased amount of PIP3*” as

$$\neg PTEN \wedge \neg SHIP1 \wedge PI3K \rightarrow PIP3.$$

In practical applications the logical implications have to be derived experimentally. In the example “MEK activates ERK” this is certainly easy, since the two proteins always appear together. But, of course, there exist more complex interactions for which the implication formula is not easily derived by common sense. For these cases we propose the following procedure:

1. Translate the experimental results into a truth table, i.e. for every stimulation tested open a new row in the table and list 0 and 1 for the absent/inhibited and present/activated components involved in the experiment.
2. Assume a logically consistent behavior and anticipate further truth values if possible.
3. Extract the canonical logical formula F as a disjunction of conjunctions. This is: one row is represented as the conjunction of all its variables. The variables occur negated if the corresponding value is 0. The subsequent formula displays a row in a truth table and its corresponding canonical formula.

$$\begin{bmatrix} A & B & C \\ 1 & 0 & 1 \end{bmatrix} \rightarrow A \wedge \neg B \wedge C.$$

The formula evolves as the disjunction of all rows.

4. Simplify F utilizing the *Quine-McCluskey* Method (30) and transform F to an implication formula.

The Quine-McCluskey Method simplifies logical formulas F obtained by truth tables to its smallest equivalent in disjunctive normal form. First, the prime implicants are identified, i.e. those implicants that have minimal length and imply F . The general idea is to apply the following fact pairwise to the non-prime implicants:

$$(x \wedge y) \vee (x \wedge \neg y) \approx x \wedge (y \vee \neg y) \approx x. \quad (1)$$

The redundant implicants are deleted in each iteration. All prime implicants are generated by this iterative procedure, but not all are needed to describe the formula F . Rather only a minimal subset of prime implicants such that all cases where F becomes TRUE are covered, is necessary. Therefore a *set covering problem* is to be solved, which is generally hard. We illustrate the procedure with the example in Figure 2.

Table 3. Truth table corresponding to blots in Figure 2.

MALT1	BCL10	CARD11	NF-kappaB
<i>measured:</i>			
0	1	1	0
1	0	1	0
1	1	0	0
1	1	1	1
<i>derived:</i>			
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	0

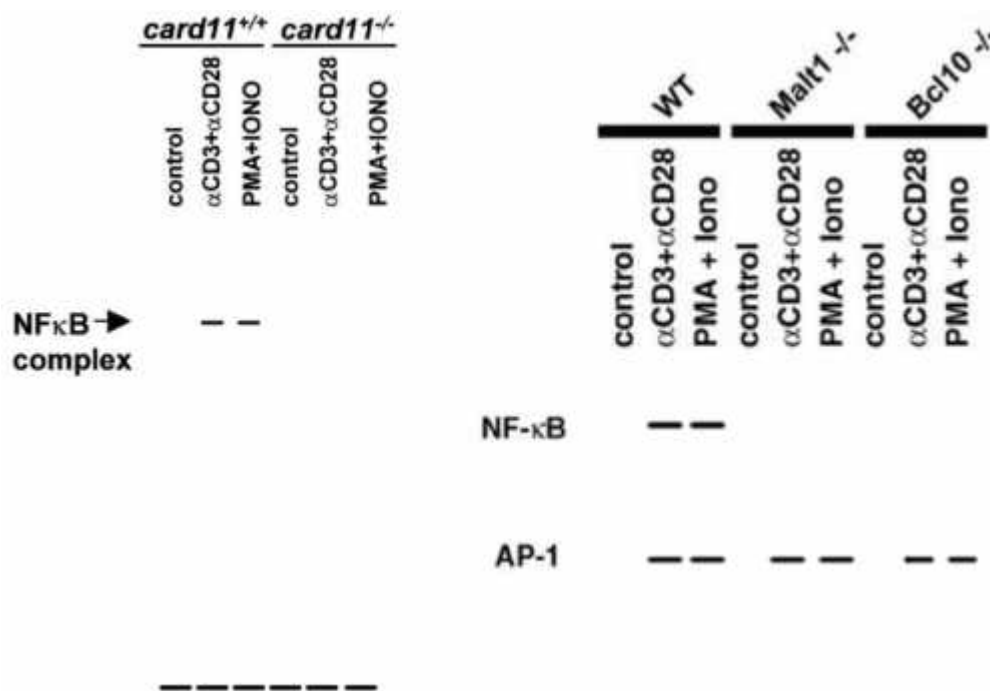


Figure 2. Two stylized blots presenting results of (31) and (32) which show the activation of NF-kappaB.

Example 2. Assume we are faced with the experimental results displayed in Figure 2. The measurements are translated into a truth table, displayed as Table 3, by introducing one row for every distinct measurement with one column for each measured component. The first four rows in the Table 3 are translated from Figure 2. For instance, in the wild type case all three components MALT1, BCL10, and CARD11 are active and we see an active NF-kappaB. This yields the row filled with 1s only. The last four rows are derived from the first four measured rows, i.e. assuming a logical implication formula we can determine the value of NF-kappaB for the missing patterns of the inputs. We know from the measurements that NF- B is active if all three inputs are present after stimulation, but inactive if only one of them is missing. Therefore, none of them has an inhibitory effect and we can conclude that NF-kappaB is also inactive if any combination of inputs is missing. Thus, we obtain a logical formula where we abbreviate MALT1, BCL10, CARD11, and NF-kappaB by M, B, C, and N, respectively.

$$F \approx (\neg M \wedge B \wedge C \wedge \neg N) \vee (M \wedge \neg B \wedge C \wedge \neg N) \vee (M \wedge B \wedge \neg C \wedge \neg N) \vee (M \wedge B \wedge C \wedge N) \vee (M \wedge \neg B \wedge \neg C \wedge \neg N) \vee (\neg M \wedge B \wedge \neg C \wedge \neg N) \vee (\neg M \wedge \neg B \wedge C \wedge \neg N) \vee (\neg M \wedge \neg B \wedge \neg C \wedge N)$$

Applying the idea of Equation (1) to each pair of implicants that are possible to be combined yields

$$F \approx (M \wedge B \wedge C \wedge N) \vee (\neg M \wedge B \wedge \neg N) \vee (\neg M \wedge C \wedge \neg N) \vee (M \wedge \neg B \wedge \neg N) \vee (M \wedge \neg C \wedge \neg N) \vee (\neg B \wedge C \wedge \neg N) \vee (B \wedge \neg C \wedge \neg N) \vee (\neg M \wedge \neg B \wedge \neg N) \vee (\neg M \wedge \neg C \wedge \neg N) \vee (\neg B \wedge \neg C \wedge \neg N) \approx (M \wedge B \wedge C \wedge N) \vee (\neg M \wedge \neg N) \vee (\neg B \wedge \neg N) \vee (\neg C \wedge \neg N)$$

This yields not only the implication, but the equivalence:

$$MALT1 \wedge BCL10 \wedge CARD11 \leftrightarrow NF\text{-kappaB}$$

stating that NF-kappaB is active if and only if MALT1, BCL10, and CARD11 are all active. If one wants to stress the necessity of a stimulus to initialize the cascade, it can be added as an additional requirement for the activeness of NF-kappaB.

Table 4 gives an overview of possible implications that occur in practical applications. Recall that activeness in this context can have any meaning you assign, e.g. phosphorylation, presence, transport to the ‘right’ location, etc.

Table 4. List of possible implications in practice. TCRB denotes the TCR bound to a ligand while TCRP and IL2P denote the phosphorylated receptors. More information on the first five examples can be found in (29) and (33) while the last one is published in (34)

statement	implication formula	explanation
ZAP70 → LAT	$A \rightarrow B$	if A is active, B is active.
FYN ∨ LCK → ABL	$A \vee B \rightarrow C$	if A , B , or both are active, C is active.
DAG ∧ VAV1 ∧ PDK1 → PKCTH	$A \wedge B \wedge C \rightarrow D$	if A , B , and C are simultaneously active, D is active.
(FYN ∧ TCRB) ∨ (LCKP1 ∧ TCRB) → TCRP	$(A \wedge B) \vee (C \wedge D) \rightarrow E$	activeness of A and B or C and D (or all) implies active E .
MKK4 ∨ MEKK1 ∨ (IL2P ∧ PI3K) → JNK	$A \vee B \vee (C \wedge D) \rightarrow E$	if A or B , or the two components C and D (or subsets of these) are active, E is active.
CXCR4 ∧ CXCL12 → -ERM	$\neg A \vee \neg B \rightarrow C$	if both A and B are present, there is no C : Absence of one of them, makes C active.

The set of formulas we obtain now describes all currently known information about the system, which we can then use to analyze and predict its behavior. However in order to do so, it proves useful to specify that we are only interested in statements that can be logically deduced from those that we have collected, and that what cannot be deduced should be considered false. This also avoids the paradoxical situation that the mathematically correct logical statement ‘FALSE → TRUE’ can lead to statements along the lines of ‘Receptor off → Receptor cascade activation’. This is known as a *closed world assumption*, see (35) (and also (36) for justification on the semantic level).

The way to formalize the above deduction rule is as follows. After computing and collecting all such implications we aggregate all formulas with an identical right-hand-side. Following this we introduce the reverse implication for each formula. In other words we assume that whenever a component on the right-hand-side is activated or inhibited there must be an explanation for it in the set of interactions. If there is not, the model must lack information or include incorrect data. This leads to the two basic interaction rules

$$A_1 \vee \dots \vee A_n \leftrightarrow B \quad (2)$$

stating that B is active if and only if at least one of the A_i is active, and

$$A_1 \wedge \dots \wedge A_n \leftrightarrow B \quad (3)$$

where B is active if and only if all A_i are active. AND-formulas (3) do occur in practical applications, but for the mathematical modeling they are obsolete as each AND-formula can be transformed to a logically equivalent OR-formula (2):

$$A_1 \wedge \dots \wedge A_n \leftrightarrow B \approx (\neg A_1 \vee \dots \vee \neg A_n \vee B) \wedge ((A_1 \wedge \dots \wedge A_n) \vee \neg B) \approx \neg A_1 \vee \dots \vee \neg A_n \vee B$$

Therefore we restrict the analysis to the specifically structured *IFF formulas* S_j .

$$S_j := \bigvee_{i \in L_j} A_i \leftrightarrow B_j \quad (\text{IFF})$$

in which a subset L_j of the components $A_1, \dots, A_n \in \{0,1\}$ determine the state of the component $B_j \in \{0,1\}$. It is assumed that the component B_j does not appear on the left-hand-side, i.e. no autoactivation takes place. The IFF-formulas do not contain AND-connection but those can always be represented by OR-formulas using artificial variables, e.g.

$$A_1 \vee A_2 \vee (A_3 \wedge A_4) \leftrightarrow B \approx (A_1 \vee A_2 \vee C \leftrightarrow B) \wedge (\neg A_3 \vee \neg A_4 \leftrightarrow \neg C).$$

Similarly, effectors (right-hand-sides) that are connected by a logical AND are split into separate IFF-formulas. In case, there is an OR-connection on the right-hand-side, an auxiliary implication is introduced using the artificial variable C :

$$A_1 \vee A_2 \vee A_3 \leftrightarrow B_1 \vee B_2 \vee B_3 \vee B_4 \approx (A_1 \vee A_2 \vee A_3 \leftrightarrow C) \wedge (C \leftrightarrow B_1 \vee B_2 \vee B_3 \vee B_4).$$

Given the IFF-formulas for interactions in the biological unit the local interdependencies are well described. By asking all IFF-formulas to be valid at the same time the individual interactions induce a global behavior of the system. By means of this observation we define the standard form of an *interaction network* as

$$\text{SN} := \bigwedge_{j=1}^k S_j, \quad (4)$$

where k denotes the number of IFF-formulas S_j which describe all relevant interactions in the system. This definition allows for a categorization of the variables: Variables that occur only on the left-hand-sides of IFF-formulas S_j are called *input* variables, while those being only in the right-hand-side of formulas are denoted *output* variables. The remaining variables are called *intermediates*. Inputs, for instance, can be receptors, environmental

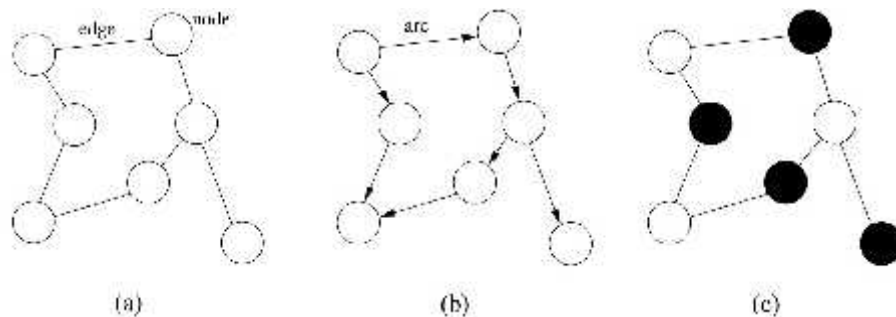


Figure 3. Visual representation of a graph, a directed graph or digraph, and a bipartite graph showing the two node partitions in white and black, respectively.

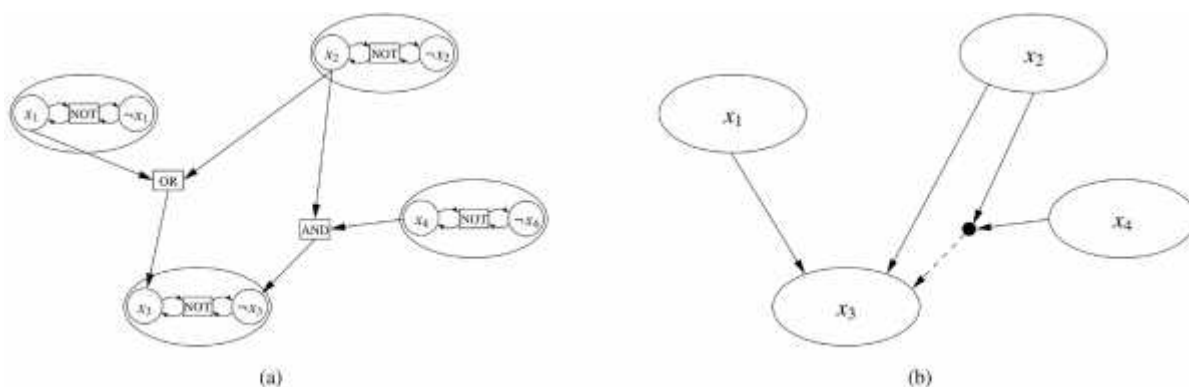


Figure 4. A small example of an interaction network with 4 components and 2 interactions, namely $x_1 \vee x_2 \leftrightarrow x_3$ and $x_2 \wedge x_4 \leftrightarrow \neg x_3$. The panel on the right-hand side shows a simplified illustrations omitting explicit OR and NOT operational nodes.

influences, or a virus in case of a disease model. Outputs may refer to transcription factors, genes, cellular phenotypes, or symptoms of a disease while the intermediates can be signaling proteins connecting receptors and transcription factors, genes that regulate other genes, etc.

With this formal model at hand we follow two main lines of analysis. We start with the investigation of structural properties of the interaction network using graph theoretical methods. In the subsequent section we explicitly go through known algorithms, show how they can be applied to an interaction network, and in particular what their use for the analysis is. In the following Section 3.3 the functionality of a system is simulated and analyzed, which will be the focus of this work. A methodology to evaluate the standard form (4) is presented, some results on the complexity of the algorithms are given, and more important for users, benefits for the analysis are demonstrated by examples.

3.2. Structural analysis

The graphical representation of biological systems is quite intuitive. Components are drawn as nodes that are connected by arrows to describe the interactions.

Graphs are not only useful for visualization, but also for structural analysis. Questions like ‘which component influences which’ and ‘are there positive or negative feedbacks involved’, i.e. does the network possess cycles with an even or odd number of negations, respectively, can be answered directly. Therefore, we first review some standard notation and formalize the approach. A *graph* (Figure 3 (a)) consists of a set of *nodes* and *edges* that are defined as pairs of nodes. If the pairs of nodes are ordered the edge becomes an *arc* and the graph becomes a *directed graph* or *digraph* (Figure 3 (b)). A graph or digraph can be *bipartite*, that is the nodes can be partitioned into two subsets such that each edge or arc has exactly one node from each subset (the white and black nodes in Figure 3 (c)).

An *interaction network* can be formally defined as a bipartite digraph with two types of nodes: components S (e.g. proteins) and operations O which contain AND, OR, and NOT to describe molecular interactions. A vertex is a combination of a component, the component’s negation and a NOT operation (Figure 4: The vertex also contains four arcs connecting the components and the NOT-operation bidirectionally. Furthermore, the biological interactions/mechanisms are encoded by arcs from

Discrete, qualitative models of interaction networks

components to operations (the requirements for an activation) and one or more arcs from an operation to component (s) (the effector(s) respectively (in)activated component(s)). The graph displayed in Figure 4 (a) depicts the formal definition of an interaction network. For simplification of illustrations we will nevertheless label a vertex simply by the component's name omitting the construct that it contains and remove the operational nodes OR. Instead we draw a direct arc from input to output and use dashed lines to denote negative influence. Furthermore the AND operation will be depicted by a small black dot. This simplified version is illustrated in Figure 4 (b).

Please note, that certain algorithms presented hereafter can actually fail if there are AND connections involved as the interaction is not independent of the values of other variables. Up to now the analysis of those cases was accomplished by the concept of hypergraphs (37). But as presented earlier in this section, we can transform each interaction network into a standard form that consists only of OR operations, which ensures the functionality of the algorithms without constructing the hypergraphs. Hence, every interaction network can be transformed into one that follows the simplified scheme of Figure 4 (b) without "black dots".

Given the interaction network as a bipartite digraph, we can apply various algorithms known from graph theory and combinatorial optimization. See (38) for a more detailed presentation of the algorithms used. With this help we analyze some properties of the network. The first question is whether one component is connected to another. A classic application of this question is that one can find out which drug targets have any influence on other components, a key question of interest in the development of pharmaceuticals. In mathematical terms the former question and checking the validity of the network, in the sense that one knows the influence of one molecule upon another and wants to verify if this connection is modeled, are the same problem. It means we have to find out if there exists a path between two nodes. We can solve it by applying the *Dijkstra algorithm*, an algorithm determining the shortest path between two nodes in a digraph with only positive weights of arcs. As long as a path exists, the algorithm provides us in polynomial time with values that specify the minimal length of the path between every pair of nodes in the graph. Otherwise it states that none exists.

It is also of interest to enumerate all paths between two given nodes. This helps to find out which components are particularly vulnerable to dysfunctions in the system. If one component is connected to the rest of the network by a small number of paths, it is more probable that this component gets disconnected than a component which has a greater number leading to this node in the graph, since its activation is redundant. On the other side, a well connected component (hub) that is malfunctioning effects a bigger set of components than a poorly connected component. Thus, it is possible to identify potential failure modes and bottlenecks of the interaction network. The enumeration of paths between two nodes s and t can be done by applying the *Dijkstra algorithm recursively*.

Firstly, calculate the shortest path between s and t . Secondly, eliminate the last arc of this path which leads directly into t . Thus the algorithm finds the "next" shortest path. This is repeated until all directed arcs into t are removed. Now move one step up the paths found so far which means remove their last arc, i.e. arcs coming from immediate predecessors of t . Iterate until the whole network is analyzed. It takes $\# \{n | n \text{ is the number of the nodes in the longest path between } s \text{ and } t\}$ recursions until all paths are listed resulting in a pseudo polynomial algorithm, as the output is obtained in polynomial time, but the output size can be of exponential size in the input length.

One important feature of interaction networks is the presence or absence of feedback loops, i.e. cycles in the graph through which a component can influence itself. Feedback loops make the analysis of networks much harder, as there can be effects occurring at a later time and therefore change the output pattern of the network. To model the interaction network at one time point it can be necessary to remove certain arcs in order to obtain a network free of cycles. For this problem we can make use of an algorithm by Tarjan (39) that enumerates all directed cycles in a digraph. It is a backtracking procedure with a running time in $O((V+A)(C+1))$, where V is the number of nodes, A the number of arcs, and C the number of directed cycles of the digraph. Therefore, the algorithm is pseudo-polynomial due to a possibly exponential number of directed cycles. Note that the trivial cycles occurring due to the negation operations are ignored (Figure 4 (a)). Therefore we can find all cycles in interaction networks. Instead of enumerating all cycles, which can be a lengthy procedure, one can also ask only for the existence of a negative or a positive feedback. For the understanding of the function of a biological system it is essential to be aware of enhancing effects or switch-off mechanisms, which makes it an important question. This can be accomplished in polynomial time. Identifying a negative or positive feedback is done by detecting the sign of weighted cycles in a digraph (40), i.e. counting the number of positive and negative influence along a closed path to determine whether its net effect is positive or negative, which can be done in polynomial time (41). However, deciding whether there is a positive or negative cycle through a fixed node v is already NP-complete (42).

Another interesting point of the analysis of interaction networks can be the manipulation of the network in order to cut the connection between some components. If one molecule has a dysfunction, it is useful to know how to separate it from the rest of the network. Let s and t be the components to be disconnected, i.e. they are the source and the sink respectively. The *Ford-Fulkerson algorithm* is applied from s to t . The algorithm computes the maximal flow between s and t in the digraph or in any subgraph. The max-flow-min-cut theorem (9) tells us that the maximal value of flow is equal to the minimal cut. In other words the minimal number of arcs which have to be removed is computable. For example, if t is connected by five direct interactions but all of them depend on the same

Discrete, qualitative models of interaction networks

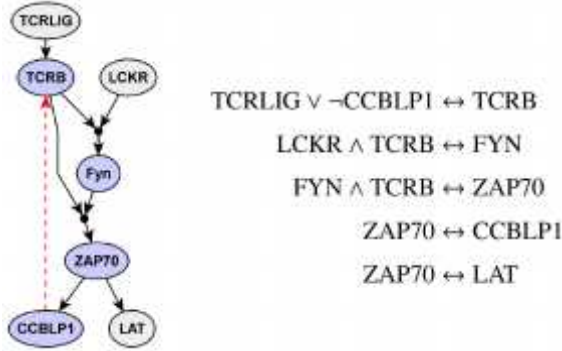


Figure 5. A small interaction network from Example 3. The dotted line denotes inhibition while the black node means a logical AND.

node at some point along their paths between s and t then flow is limited by this single node, i.e. it is a bottleneck in the system. Linear programming duality also provides us with the specific arcs which have to be removed which means we can either target a bottleneck node or a specific interaction coming or originating from that node. The latter may allow for a better intervention control in case one wants to suppress undesired behavior such as it is the case in disease models or drug targeting. For example the protein Ras is often involved in cancer development, yet being a hub it controls so many growth processes that the toxic effects are a problem (43). Instead of hitting a hub causing unwanted side effects, a more restricted set of changes downstream of it may be beneficial.

As displayed above, we can identify and eliminate feedback loops. Neglecting the trivial cycles occurring due to the negation, every interaction network can be converted into an acyclic digraph. In this context we can layer our graph, i.e. we can clearly assign a number to every node which refers to the level it belongs to. An interaction network is *layered* when we can label the nodes such that the component nodes have an odd layer number, all operation nodes have an even layer number, and the orientation of each arc follows the increasing layer number. The layer number provides a hint at the position of a component in the network. It also yields the number of components that are upstream via the longest activation cascade. We note that it is sufficient for an interaction network to contain no cycles to be a layered network. This can be proven by a *Breadth First Search Traversal* type of algorithm displayed in (44), that layers any acyclic interaction network. The idea of the algorithm is to assign the next layer number to the successors of the considered layer. In case a successor has only one incoming arc or all its predecessors are in the previous layer, the node is assigned the next integral number. Otherwise, the arcs leading from the considered layer to the successor are removed and the assignment of a layer number is postponed. With this we close the illustration of the potentials of the structural analysis for interaction networks and move to the functional part.

3.3. Functional analysis

The biological system is syntactically characterized by its standard form SN. In order to analyze

its semantics and thus its functionality concerning activation/inhibition behavior an instrument to evaluate the logical IFF-formulas is needed. Therefore, we exploit the concept of *satisfiability*, which asks for a 0/1 assignment to the components such that the overall function evaluates to TRUE (see (15) for a formal definition), and introduce this special case:

The IFFSAT problem of a problem SN with literal set L : We fix the value of a set of components $L_0 \subseteq L$ to be 0 and another set $L_1 \subseteq L$ of components that are 1. Then the problem is to decide whether a solution for the following formula exists or not, i.e. whether

$$SN \wedge \bigwedge_{x \in L_0} (\neg x) \wedge \bigwedge_{x \in L_1} x \quad (\text{IFFSAT})$$

is satisfiable and if so, give an appropriate assignment.

In the setting of IFFSAT problems we can answer the question whether, given a partial set of activations, there exists a solution for the entire formula S , by fixing some logical variables in S to a prescribed value and solving the IFFSAT problem for the remaining formula S' . In biological applications this general procedure yields the prediction of a systems response to certain stimuli and dysfunctions, the detection of corresponding intervention strategies, and model verification.

Example 3. We consider the interaction network of Figure 5 representing a subnetwork of the TCR model of (29): A key issue for modeling biological systems is to check the completeness and correctness of the model. This can be accomplished by checking whether IFFSAT is inconsistent, i.e. there exists no satisfying truth assignment, for a given set of measured data. In our example interaction network (Figure 5) a measurement that states the simultaneous activation of TCRLIG, CCBLP1, and TCRB leads to an inconsistent IFFSAT formula as it contradicts the IFF-formula $TCRLIG \wedge \neg CCBLP1 \leftrightarrow TCRB$. Thus, there must exist another component that activates TCRB and is not present in the interaction network yet, or a temporal information is lacking. On the other hand, the measurement $TCRB = 1$ and $LAT = 1$ is consistent with the model SN.

Several scenarios can be tested with these logical IFF-formulas. First of all, certain input and output patterns can be checked for validity. For this purpose fix TCRLIG, LCKR, and LAT to the desired value and solve the related IFFSAT instance. If it is satisfiable, the input/output pattern is a valid assignment. If one is interested in predicting the output for a prescribed input pattern, one fixes the inputs to interesting values again and solves IFFSAT twice: Once with value $LAT = 0$ and one time with $LAT = 1$. If both situations are satisfiable, LAT is undetermined as it is the case, e.g. for $TCRLIG = 1$, leaving LCKR arbitrary. This happens typically when there is insufficient information about the activation of a node which in signaling networks is the case for *horizontal inputs*. These inputs are critical to uniquely determine the state of a molecule after receptor

Discrete, qualitative models of interaction networks

triggering, but their own regulation is unknown as for instance they are controlled by other receptors or undetermined environmental variables. In case one instance is satisfiable and the other is not, LAT takes the value obtained in the satisfiable setting, which is true for the input pattern TCRLIG=1 and LCKR=1 implying LAT=1.

Failure modes are crucial to the development of pharmaceuticals. In mathematical terms failure modes are simply certain values of variables that lead to an undesired behavior of the biological system. The corresponding intervention strategies can be obtained by fixing the failure modes to their opposite “good” value and computing whether other variables are forced to obtain a specific value, which then are possible intervention strategies. As in the previous tests we solve several different IFFSAT instances to find fixations. In our example we set ZAP70=1, which is a key component in the TCR signaling network, and ask for variables that force this setting. In this case a possible intervention is FYN=1 and TCRB=1 as well as TCRLIG=1 and LCKR=1.

Much research has been done to find effective solution algorithms for subclasses of SAT (45). There is, however, no algorithm that can efficiently check satisfiability for arbitrary propositional formulas (46). Nevertheless, a great deal of successful research has been performed to develop practically usable methods for huge SAT problem instances. Note that in general IFF-formulas cannot be transformed into a series of formulas obtaining a certain “nice” structure that allows for polynomial computations, e.g. 2SAT or Horn formulas (15). Thus, generally IFFSAT cannot easily be solved by using the Prolog (47) method. Actually, IFFSAT is equally hard as general SAT:

Lemma 1. *The satisfiability problem IFFSAT is equivalent to 3-SAT, hence NP-complete.*

For the proof see (48). Nevertheless, IFFSAT becomes much easier if a certain structural property is fulfilled as presented in (49). We will omit the technical details of these structural properties and state only their main result:

Theorem 1. *If an SN instance in cascade form satisfies the cutnode condition, the related IFFSAT can be solved in linear time.*

Note that each SN instance can be transformed into this cascade form but not every instance fulfills the cutnode condition. However, the often problematic feedback loops are not the reason for an unsatisfied cutnode condition. For the details see (49).

Another method that is equivalent to IFFSAT is an integer programming based approach. Instead of utilizing SAT for the semantics of SN, we exploit a feasibility or optimization problem over a linear system of inequalities requiring integrality of the variables. An

overview of general integer programming theory can be found in (50).

Dantzig (51) already presented how a SAT problem can be formulated as an integer program (IP). For the IFFSAT problem the associated integer program is constructed by introducing $|L|$ binary variables x_i and their complements \bar{x}_i , and translating each IFF-formula into the system

$$\begin{aligned} \sum_{i \in L_j} x_{A_i} - x_{B_j} &\geq 0 && \text{for } S_j, j=1, \dots, k \text{ all} \\ -x_{A_i} + x_{B_j} &\geq 0 && \text{for all } i \in L_j \text{ and} \\ &&& S_j, j=1, \dots, k \end{aligned} \quad (5)$$

$$x_l + \bar{x}_l = 1, \quad x_l \in \{0, 1\} \quad l \in L$$

$$x_p = 1, \quad x_q = 0 \quad p \in L_1, q \in L_0$$

where we will assume that for non-negated components $A \in L$ the variable x_A , and for negated components $\neg A \in L$, \bar{x}_A has been used in the formulation of the inequalities. The first inequality simply states that B is 0 if all inputs A_i are zero while the second type of inequalities force B to be 1 in case any of the A_i is active. The condition $x_l + \bar{x}_l = 1$ requires that the value of a component is either 0 or 1 but never both at the same time.

The feasible solutions of the IP correspond to an assignment of 0/1-values to each component that satisfies all restrictions. As in the IFFSAT setting the considered biologically relevant questions can be answered, partially with less instances to solve, as we can apply an objective function.

However, solving integer programs is generally hard, as the related decision problem is NP-complete. Thus, it is not expected that it is more efficient than the SAT based approach. In some cases it can be solved easily due to combinatorial features or special descriptions of the underlying polyhedron. The following lemma yields one easy case in which the description of the polyhedron enables us to solve the corresponding IFFSAT instance in polynomial time, this is the case for a single IFF formula of arbitrary (but polynomially bounded) size.

Lemma 2. *The inequality description of a single IFF formula S_j in the IP model (5) is integral.*

See (52, p. 338) for the proof. Nevertheless, it cannot be generalized to IFFSAT problems with an arbitrary number of equivalence formulas, as the property that permitted fast computations is not preserved.

Computations utilizing both approaches still perform in a reasonable time. In Table 6 we used the TCR

Discrete, qualitative models of interaction networks

Table 5. Structural facts about the TCR interaction network from (29).

# Comp	# inter	min $ L_j $	max $ L_j $	avg $ L_j $	# inputs	# outputs	# paths: in/out	Shortest path	Longest path
93	123	0	6	1.32	3	14	5630	9	60

Table 6. Computational performance of the IP method by means of the TCR model of (29).

Variables	Rows	Inputs	Outputs	#feas	#infeas	Total time (s)	Avg time (s)
214	376	3	14	36	131036	≈ 120	0.001

model introduced in (29) to test the computational performance of the IP method. Structural facts about the interaction network are given in Table 5. All possible input/output patterns are fixed and tested for feasibility. The average computational time of 0.001 seconds per instance promises an effective method to test the various biological questions. Computations were performed on a SUN-Fire V890 (1.2GHz) using CPLEX 9.1 (53).

4. DYNAMIC MODEL

In practical applications IFFSAT instances often turn out to be infeasible even without components fixed. It occurs that there are errors in the interactions, especially when the model is developed step by step or in multiple units. Mistakes are then often made when merging the parts. Another source for infeasibility is that temporal information is not represented in the SN model. If it is in fact available, which in practice is often not the case, it can lead to infeasibility. In particular, delayed interactions modeled as instantaneous are a risk, as for instance (negative) feedback loops can lead to an inconsistent IFFSAT, but at the same time have a huge impact on the functionality of a signaling network through their timing since certain activation cascades can be enabled initially and switched off at a later time point to avoid over-reaction. So far, the detection of errors is usually performed by experts who validate the interaction network step by step in order to find the crucial point for infeasibility. In this section we introduce a new method to systematically investigate and propose ‘repair’ options of causes for infeasibility in SN which, as a side effect, allows the handling of timing information. A typical situation is illustrated in the example.

Example 4. While merging two separate models it happens that two different pathways lead to the same common component possibly generating a conflict. One pathway activates the component while the other does not. In (33) the TCR and IL2 receptor of T cells are integrated. In this process several common components occurred. For example, the secondary messenger DAG is produced following stimulation of either receptor and can activate PKCs and RASGRP. However, it is known that RASGRP does not become activated upon IL2 stimulation. The model predicts RASGRP to be active while experimental results state the opposite. Thus, we fix RASGRP to be 0, obtain therefore an infeasible configuration, and compute the minimal infeasible subsystems which help to detect the crucial point in the highly complex overall system. In this case, there is only one minimal infeasible subsystem which simplifies the analysis. This is displayed in Figure 6. Here, we could reveal that the source of DAG might be critical

due to its different location and/or additional - but so far unknown - regulatory factors. In general, infeasibilities might not be as easy to unravel. Stimulating CD28 in the merged TCR and IL2 model of (33) and requesting JNK to be active yields 771 minimal infeasible subsystems. Several of them are almost equivalent, so a classification has to be done to make this set understandable. This is analogous to the case where a certain set of interactions can be connected by many paths to construct feedback loops, making the complete set of feedbacks large, with a concentrated core of interactions that explain most or all of the infeasible subsystems.

When the infeasibilities did not arise due to modeling mistakes or as a conflict in the data that is resolved by experiments it probably point to delayed interactions and an extension to the standard logical description of the network is required. Suppose we are given the standard form SN of an interaction network. For each IFF formula $S_j, j=1, \dots, k$, we introduce auxiliary variables $y_i, i \in L_j$ indicating the influence of Variable A_i on B_j . This leads us to the *requirement IFF formulas*:

$$R_j = \bigwedge_{i \in L_j} \left(\neg(A_i \vee \neg y_i \vee B_j) \wedge \bigvee_{i \in L_j} A_i \vee (\neg y_i \vee \neg B_j) \right) \wedge \left(\bigvee_{i \in L_j} (A_i \wedge y_i) \rightarrow B_j \right) \wedge \left(\bigvee_{i \in L_j} (B_j \wedge y_i) \rightarrow \bigvee_{i \in L_j} A_i \right) \quad (\text{RIFF})$$

which assigns to each interaction (IFF formula) S_j *requirement variables* $y_i, i \in L_j$. The RIFF formula states on the one hand that B_j is active if there is an $i \in L_j$ such that both A_i and its requirement variable y_i are 1 -- causes have consequences -- and on the other hand if B_j and some y_i are 1, then there must be an A_i that is active, i.e. there is no consequence without cause. Concretely, in an interaction network $y_i=1$ denotes influence of the corresponding component A_i on the right hand side B_j while $y_i=0$ denotes no effect. In case $y_i=0$ for all $i \in L_j, B_j$ is requested to be free. The second condition is needed for a monotonic behavior of the y variables which is needed for further computations. For details see (48). Analogously we define the *RIFFSAT problem*.

$$\bigwedge_{j=1}^k R_j \wedge \bigwedge_{x \in L_0} (\neg x) \wedge \bigwedge_{x \in L_1} x \quad (\text{RIFFSAT})$$

with L_0 describing the set of components that shall be inactive and L_1 the set of active components.



Figure 6. Minimal infeasible subsystem resolving one conflict while merging the two receptor models in (33).

We demand that the requirement variable y_t of A_i are different for every IFF formula in which A_i occurs on the left hand side in order to deal with a component occurring in several interactions separately in each situation. Note, that in the graphical representation the requirement variables can be seen as binary variables for the arcs that indicate whether the arc is present or not. Therefore, one can represent the dynamic behavior of a network by introducing T copies of the same interaction network each associated with a binary vector $y_t, t = 1, \dots, T$ that indicates which interaction is present at time t . The difference $y_t - y_{t-1}$ encodes the change in structure of the system from time $t-1$ to t .

Example 5. In the merged TCR and IL2 network (33) several interactions occur at later points in time because of long production times or dependency on components that are not produced/degraded at an early state. Examples of these are the clauses “ERK absence and LCKP1 activity implies the activity of SHP1”, “activity of GADS, LAT, and ZAP70 implies activity of GAB2”, and similarly “If GRB2, ZAP70, and LAT are active, so is GAB2”. Modeling the networks behavior including these different points in time reveals, for instance, that ERK is active after simultaneous TCR and IL2 stimulation while it remains inactive if the cell is prestimulated at IL2 and subsequently also at the TCR. This was experimentally confirmed (33).

As for IFFSAT we can represent the set of all truth assignments for which (RIFFSAT) is TRUE by an inequality description: For every RIFF formula R_j and corresponding requirement set Y_j we write down inequalities of the form

$$\begin{aligned} \sum_{k \in L_j} x_{A_k} + (1 - y_i) &\geq x_{B_j} && \text{for all } i \in L_j \\ x_{A_i} - (1 - y_i) &\leq x_{B_j} && \text{for all } i \in L_j \\ x_l + \bar{x}_l &= 1, \quad x_l \in \{0, 1\} && l \in L_j \cup \{B_j\} \cup Y_j \\ x_p &= 1, \quad x_q = 0 && p \in L_1, q \in L_0 \end{aligned} \quad (6)$$

We will assume that \bar{x}_i is used in the inequality description if the corresponding component is negated. Note that for each fixed y the formulation reduces to an IFFSAT instance.

A satisfying truth assignment yields again an assignment of 0 and 1 to all components of the unit, but it

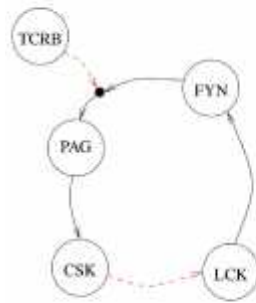
might not be valid for the related IFFSAT, as we can cut the connection between certain components. In particular, if the original IFFSAT problem is inconsistent, a solution to the associated RIFFSAT implies which interactions of components have to be removed in order for IFFSAT to be consistent, i.e. the value of variables A_i in S_j for which $y_i = 0$ have to be irrelevant for B_j . This points at the origin of the inconsistencies. In order to identify the location of a modeling error, one is usually interested in minimal inconsistent or maximally consistent solutions to RIFFSAT. The optimality refers to the number of zeros in y as this encodes the variables to be removed from the IFF formula.

Both approaches, RIFFSAT as well as IP (6), can provide one maximal feasible solution. But one is of course interested in all maximal solutions with respect to the requirement variables in order to find all possible modeling errors, missed temporal information, or to plan specific experiments verifying an actual network structure. In order to find the set of all maximal solutions with respect to the requirement variables it is necessary to use a ‘clever enumeration method’. For this purpose, we make use of the joint generation algorithm (54). For an application to signaling networks and its computational performance see (48). The joint generation algorithm does not only provide a maximally consistent subnetwork, but the minimally inconsistent subsystems are generated as a by-product which in particular yields a concrete characterization of inconsistencies. This allows for an easy identification of modeling errors. The subsequent example illustrates the method.

Example 6. Consider the negative feedback that is depicted in Figure 7. Leaving all variables unfixed, the corresponding SN is feasible with the assignment $\{x_{\text{TCRB}} = 1, x_{\text{FYN}} = 1, x_{\text{PAG}} = 0, x_{\text{CSK}} = 0, x_{\text{LCK}} = 1\}$. If we assume that TCRB is inactive, i.e. $x_{\text{TCRB}} = 0$, the situation changes and the system becomes infeasible. Generating all maximal solutions of the IP presented in Figure 7 with the objective to maximize the number of attending implications, yields in particular the minimal infeasible subsystem which is the ‘cycle’ comprised of the components FYN, PAG, CSK, and LCK. This implies that either a modeling error occurred while composing different parts to one overall system, or the interactions occur at different times. In this setting as all reactions are experimentally proven, so that rather the second reason applies. In (29) it is presented that FYN phosphorylates PAG after 3 to 5 minutes which is ‘late’ compared to the other implications considered. For TCR signaling this interaction takes the role of an off-switch.

5. MULTIPLE ACTIVATION LEVELS

The propositional logic approach proved to be a useful framework for the analysis of large interaction networks. Functionality and structure can be investigated hand in hand. In particular for biological systems like signal transduction networks which often lack quantitative



$$\begin{aligned}
 (1 - x_{PAG}) &\geq x_{TCRB} - (1 - y_{TCRB}) \\
 (1 - x_{PAG}) &\geq (1 - x_{FYN}) - (1 - y_{FYN}) \\
 (1 - x_{PAG}) &\leq (1 - x_{FYN}) + x_{TCRB} + (1 - y_{TCRB}) \\
 (1 - x_{PAG}) &\leq (1 - x_{FYN}) + x_{TCRB} + (1 - y_{FYN}) \\
 x_{CSK} &\geq x_{PAG} - (1 - y_{PAG}) \\
 x_{CSK} &\leq x_{PAG} + (1 - y_{PAG}) \\
 x_{LCK} &\geq (1 - x_{CSK}) - (1 - y_{CSK}) \\
 x_{LCK} &\leq (1 - x_{CSK}) + (1 - y_{CSK}) \\
 x_{FYN} &\geq x_{LCK} - (1 - y_{LCK}) \\
 x_{FYN} &\leq x_{LCK} + (1 - y_{LCK}) \\
 x_j, y_j &\in \{0, 1\} \quad \forall j \in \{TCRB, \dots, FYN\} \\
 x_{TCRB} &= 0
 \end{aligned}$$

Figure 7. A negative feedback that is in general feasible, but becomes infeasible for $TCRB=1$, and its dynamic model description.

measurements so that the data is presented in a binary form. For instance, the kinase ERK being phosphorylated or not and the transcription factor NFAT being localized to the nucleus or the cytoplasm. Thus, binary modeling is the only approach suitable. However, certain interaction networks or parts of a network may allow for a more detailed analysis. Even though the data is not yet continuous several *levels* of activation can be identified. For example, a component can be inactive, of low, medium, or high activity. Depending on its state it can have different effects on the behavior of the network. The next example illustrates this behavior in an apoptosis network.

Example 7. The apoptosis network of Schlatter *et al.* (26) is a logical model describing the process of programmed cell death. It includes components that have three distinct activation levels. For instance, the component *UV radiation* can be off, low, or high. The authors observe that a cell survives if there is no UV radiation and surprisingly if it is high, while it dies if UV radiation is low. The experimental evidence is given in Figure 8. It presents the percentage of cells that survive upon different radiation levels, which are indicated below by 0,1, and 2 referring to no, low and high UV light, respectively.

Taking a closer look at the apoptosis network we find several level dependent interactions such as “If there is no Caspase3 we find activation of Gelsolin, while if we find no high Caspase3, ICAD is activated” (see (26) and reference therein for biological details). In our usual notation this reads

$\neg \text{Caspase3} \leftrightarrow \text{Gelsolin}$ $\neg \text{high Caspase3} \leftrightarrow \text{ICAD}$
 If we assume again that all such interactions are valid at the same time such level dependent interactions yield distinct read-outs of the network.

Obviously, the different activation levels can be discretized and converted into a Boolean model. But this implies a great loss of information and thus reduces the

predictive power of the model. Alternatively, a function can be fitted through the valid points and a continuous technique can be applied, but then a large part of (unknown) information has to be assumed. It is therefore desirable to bridge the gap between the purely logic approach and continuous methods, which assumes a complete knowledge of all components’ concentrations over time. A suitable model for the purpose of including multiple activation levels has to allow a *finite* range for the variables and at the same time it has to keep its propositional logic characteristics that enables one to decide feasibility of the system or of given scenarios. An easy approach that fulfills these conditions is to introduce binary variables for each possible level of a component that indicate its value together with the constraint that exactly one of the binary variables corresponding to one component is active. If we denote the level of a component by a superscript, then the level dependent interactions of Example 7 look as follows:

$$\text{Caspase3}^{inact} \leftrightarrow \text{Gelsolin}^{act} \quad \text{and} \quad \text{Caspase3}^{inact} \vee \text{Caspase3}^{low} \leftrightarrow \text{ICAD}^{act}$$

with $\text{Caspase3}^{inact} + \text{Caspase3}^{low} + \text{Caspase3}^{high} = 1$, $\text{Gelsolin}^{inact} + \text{Gelsolin}^{act} = 1$, and $\text{ICAD}^{inact} + \text{ICAD}^{act} = 1$. The concept goes back to van Ham (20) in the 1970’s and is currently the standard way of dealing with multiple activation levels computationally (e.g. (55)) as well as theoretically (56,57,58,59). This approach is well suited for cases with a limited number of components that have only a few activation levels distinct from 0 and 1. However, as the systems become more complex, increasing components and/or the levels of activation, the system size escalates, in particular the variable number rises by approximately $\# \text{components}(\max \# \text{levels} - 1)$. It is indeed reported in Schlatter *et al.* (26) that computational time increases dramatically even when only a few components have more than two levels of activation associated. In Thomas’ concept of kinetic logic (Section 3.3) multiple levels can be incorporated in a different manner: Snuossi

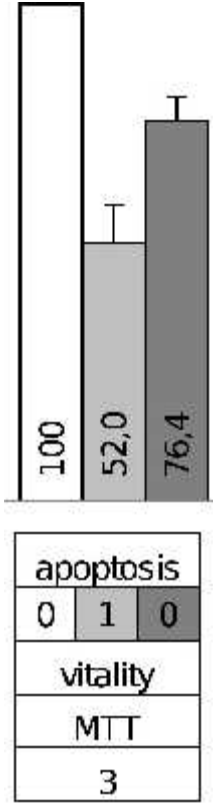


Figure 8. Experimental results showing the effect of different levels of UV radiation taken from (26). The percentage of surviving cells for untreated cells (0) and after weak UV (1) and strong UV (2) stimulation for each measure are shown in columns in the upper half including standard deviation. The ‘3’ refers to the number of repeats half including standard deviation.

(60) introduces multiple values for one variable which is increased whenever a reaction is executed again. In Thomas’ state transition graph this is straight forward to integrate although complexity of the graph rises significantly. In this framework, it is the standard way of integrating levels (see also (21)). However, it is not clear how to model this in the Boolean network approach without assuming consecutiveness of activation levels and without the use of the state transition graph which bypasses the inclusion of integer values into the logical activation formulas.

We therefore propose a different approach that avoids an explosion of the system, but keeps one variable per component. The variables become integer valued rather than binary and its value expresses its current level. No additional variables are introduced. The idea is to keep the formalism of Section 3.3, but associate with every component in each IFF formula an interval determining the components’ levels at which it has an effect in the current formula. More formally, we introduce *activation levels* $\{0,1,\dots,n_i\}$ for each component A_i, B_j . Starting with an

IFF formula S_j we write down *interval* IFF formulas (*iIFF formulas*):

$$W_j := \bigvee_{i \in L_j} (A_i \in J_{i,j}) \leftrightarrow (B_j \in J_{B_j}). \quad (\text{iIFF})$$

in which each variable $A_i, i \in L_j$ is associated with an interval $J_{i,j} \subseteq \{0,1,\dots,n_i\}$, and B_j with $J_{B_j} \subseteq \{0,\dots,n_{B_j}\}$ denoting the *activation window*, which can differ for all variables and each of their occurrence.

We call an expression $A_i \in J_{i,j}$ TRUE if A_i takes a value from $J_{i,j}$. Correspondingly, an iIFF formula states that whenever at least one of the A_i takes a value within $J_{i,j}$, then B_j attains a value of J_{B_j} , and is therefore a natural generalization of an IFF formula to more than two levels of activation. However, in contrast to IFF formulas we do not assume the variables to occur negated. We rather shift this information into the interval J . Whenever the same variable A_i occurs in several or even in one iIFF formula, different activation windows $J_{i,j}$ can thus be associated.

Coming back to our example of apoptosis (26), the interactions depending on multiple levels can be formalized as:

$$\text{Caspase3} \in \{0\} \leftrightarrow \text{Gelsolin} \in \{1\} \quad \text{and} \quad \text{Caspase3} \in \{0,1\} \leftrightarrow \text{ICAD} \in \{1\}$$

In accordance with the original IFFSAT we can define the *interval IFFSAT problem*: Decide whether there exists an assignment of integer values to each component such that

$$\bigwedge_{j=1}^k W_j \wedge \bigwedge_{x \in F} (x \in J_f) \quad (\text{iI}, \text{FF}, \text{S}, \text{A}, \text{T})$$

becomes TRUE. In this context L is the set of components in SN, $F \subseteq L$ a subset of predetermined components, and k the number of iIFF formulas.

Note that iIFFSAT, just like RIFFSAT, contains IFFSAT as a special case, in which all intervals have only one element, either 0 for negative or 1 for positive components. Therefore, iIFFSAT is as powerful for multiple level modeling as IFFSAT is for binary interaction networks. Hence, the relevant biological questions can be checked for equivalently (see Example 3). However, this *interval satisfiability* is not investigated as well as SAT is. Automatic solvers are not available. To solve iIFFSAT computationally we rewrite it as a system of nonlinear equalities. We encode feasible states as common roots of polynomial equations (61):

Discrete, qualitative models of interaction networks

For each iIFF formula W_j we collect all minimal variable assignments such that W_j is TRUE. Therefore, we obtain two different kinds of points, those with $B_j \in J_{B_j}$ and those with $B_j \notin J_{B_j}$. The first type are all points $x_i^l = (x_{A_i}^l, x_{B_j}^l) \in J_{i,j} \times J_{B_j}$, $l = 1, \dots, m_i$, which we obtain for every $i \in L_j$. The latter are o_j points $y^l \in \bar{J}_{1,j} \times \bar{J}_{2,j} \times \dots \times \bar{J}_{|L_j|,j} \times \bar{J}_{B_j}$, where $\bar{J}_{i,j}$ denotes the complement of the set $J_{i,j}$ with respect to $\{0, 1, \dots, n_i\}$. The feasible points can be encoded as the roots of a polynomial:

$$\prod_{i \in L_j} \prod_{l=1}^{m_i} \left[(A_i - x_{A_i}^l)^2 + (B_j - x_{B_j}^l)^2 \right] \cdot \prod_{i \in L_j} \left[\sum_{l=1}^{o_i} (A_i - y_{A_i}^l)^2 + (B_j - y_{B_j}^l)^2 \right] = 0.$$

Each variable x that is fixed to a prescribed interval J_f is written as a separate polynomial:

$$\prod_{v \in J_f} (x - v) = 0.$$

Further, we introduce a polynomial for each variable that is not fixed to a prescribed interval to model its possible activation windows. For each variable x with activation levels $\{0, 1, \dots, n_x\}$ we write

$$\prod_{j=0}^n (x - j) = 0$$

to express the feasible region of the variables.

To capture iFFSAT, such polynomials are introduced for each W_j , and each variable x yielding a equality system of polynomials which can be investigated for feasibility by means of Hilbert's Nullstellensatz.

Theorem 2 (see e.g. (62)). Let $p_i \in C[x_1, \dots, x_n]$, $i = 1, \dots, m$. Then:

$$\{x \in C^n \mid p_i(x) = 0 \forall i = 1, \dots, m\} = \emptyset \Leftrightarrow 1 = \sum_{i=1}^m r_i p_i \quad \text{with } r_i \in C[x_1, \dots, x_n]$$

Just as in the SAT case we have thus a certificate for the feasibility of the modeled system, which allows us to follow the same line to answer questions of the previous type concerning the prediction of a system's output, simulating its behavior, and detection of intervention strategies. We illustrate this approach using our example from apoptosis which in this formulation reads

$$\begin{aligned} 0 &= (\text{Caspase3}^2 + (\text{Gelsolin} - 1)^2) \cdot ((\text{Caspase3} - 1)^2 + \text{Gelsolin}^2) \\ 0 &= (\text{Caspase3}^2 + (\text{ICAD} - 1)^2) \cdot ((\text{Caspase3} - 1)^2 + (\text{ICAD} - 1)^2) \cdot ((\text{Caspase3} - 2)^2 + \text{ICAD}^2) \\ 0 &= \text{Caspase3} \cdot (\text{Caspase3} - 1) \cdot (\text{Caspase3} - 2) \\ 0 &= \text{Gelsolin} \cdot (\text{Gelsolin} - 1) \\ 0 &= \text{ICAD} \cdot (\text{ICAD} - 1). \end{aligned}$$

In a sense this kind of reformulation is similar to the state transition graph approach in Thomas' kinetic logic

by Snoussi (60). While in state transition graphs the network's overall states are enumerated and ordered, we enumerate all feasible points of only one interaction of the system in the polynomial equations. Doing this for every interaction guarantees the global behavior. Computational experience with this type of formulations is still lacking and an developing field of research.

6. CONCLUSIONS

Discrete models are an important and viable tool to model logical and multi-level qualitative behavior of cellular signaling networks. Their strength lies in the flexibility to integrate information at varying biological levels. They are not limited to steady state logical analysis, but can in a natural fashion encode all possible timing behaviors when only partial and relative timing information is available. Usable algorithmic tools for the analysis of the arising mathematical models are available, but are themselves an interesting topic for further research: SAT solvers like MiniSAT (63) can solve problems with hundred thousands of variables, the free implementation of the joint generation algorithm available (64) has been used successfully in realistic applications (33), and libraries of graph algorithms are available for most programming languages, e.g. the Boost library for C++ (65).

7. REFERENCES

1. James Dickson Murray. *Mathematical biology: I. An introduction*. Springer-Verlag, 1 edition (1989). Third edition available from 2002
2. Leonor Michaelis, Maud Leonora Menten. Die Kinetik der Invertinwirkung. *Biochemische Zeitschrift* 49, 333–369 (1913). In German
3. Stuart A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* 22 (3), 437–467 (1969)
4. Leon Glass, Stuart A. Kauffman. The logical analysis of continuous non-linear biochemical control networks. *J. theor. Biol.* 39 (1), 103–129 (1973)
5. Stuart A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, USA, 1 edition (1993)
6. István Albert, Juilee Thakar, Song Li, Ranran Zhang, Réka Albert. Boolean network simulations for life scientists. *Source Code for Biology and Medicine* 3 (16) (2008)
7. Afshin Esmaeili, Christian Jacob. Evolution of discrete gene regulatory models. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation, GECCO '08*, 307–314. ACM, New York, NY, USA (2008)
8. Steffen Klamt, Julio Saez-Rodriguez, Jonathan A. Lindquist, Luca Simeoni, Ernst Dieter Gilles. A methodology for the structural and functional analysis of

Discrete, qualitative models of interaction networks

signaling and regulatory networks. *BMC Bioinformatics* 7 (56) (2006)

9. Alexander Schrijver. *Combinatorial optimization. Polyhedra and efficiency*. Vol. A, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, Germany (2003). ISBN 3-540-44389-4

10. Regina Samaga, Axel von Kamp, Steffen Klamt. Computing combinatorial intervention strategies and failure modes in signaling networks. *Journal of Computational Biology* 17 (1), 39–53 (2010)

11. Carl Adam Petri. *Kommunikation mit Automaten*. 2. Schriften des Instituts für Instrumentelle Mathematik (1962). In German

12. Monika Heiner, David Gilbert, Robin Donaldson. Petri nets for systems and synthetic biology. In M. Bernardo, P. Degano, G. Zavattaro, editors, *SFM 2008*, 215–264. Springer-Verlag, Berlin, Heidelberg (2008)

13. René Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology* 42 (3), 563–585 (1973)

14. René Thomas, Richard D’Ari. *Biological Feedback*. CRC Press, Boca Raton, Florida (1990). ISBN 0-8493-6766-2

15. Hans Kleine Büning, Theodor Lettmann. Propositional Logic: *Deduction and Algorithms*, volume 48 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press (1999)

16. M. Kaufman, René Thomas. Model analysis of the bases of multistationarity in the humoral immune response. *Journal of Theoretical Biology* 129 (2), 141–162 (1987)

17. Lucas Sánchez, Denis Thieffry. A logical analysis of the *Drosophila* gap-gene system. *Journal of Theoretical Biology* 211 (2), 115–141 (2001)

18. Adrien Fauré, Aurélien Naldi, Claudine Chaouiya, Denis Thieffry. Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics* 22 (14), e124–e131 (2006)

19. Adrien Fauré, Denis Thieffry. Logical modelling of cell cycle control in eukaryotes: a comparative study. *Molecular BioSystems* 12 (5), 1569–1581 (2009)

20. P. Van Ham. How to deal with variables with more than two levels. In *Kinetic Logic*, volume 29 of *Lecture Notes in Biomathematics*, 326–343. Springer (1979)

21. Heike Siebert, Alexander Bockmayr. Incorporating time delays into the logical analysis of gene regulatory networks. In *Computational Methods in Systems Biology*, 169–183. Springer (2006)

22. Kanae Oda, Yukiko Matsuoka, Akira Funahashi, Hiroaki Kitano. A comprehensive pathway map of epidermal growth factor receptor signaling. *Molecular Systems Biology* 1 (2005.0010) (2005)

23. Song Li, Sarah M. Assmann, Réka Albert. Predicting essential components of signal transduction networks: A dynamic model of guard cell abscisic acid signaling. *PLoS Biology* 4 (10), e312 (2006)

24. Simone Gupta, Siddharth S. Bisht, Ritushree Kukreti, Sanjeev Jain, Samir K. Brahmachari. Boolean network analysis of a neurotransmitter signaling pathway. *Journal of Theoretical Biology* 244 (3), 463–469 (2007)

25. Rafal Zielinski, Pawel F. Przytycki, Jie Zheng, David Zhang, Teresa M. Przytycka, Jacek Capala. The crosstalk between EGF, IGF, and insulin cell signaling pathways - computational and experimental analysis. *BMC Systems Biology* 3 (88) (2009)

26. Rebekka Schlatter, Kathrin Schmich, Ima Avalos Vizcarra, Peter Scheurich, Thomas Sauter, Christoph Borner, Michael Ederer, Irmgard Merfort, Oliver Sawodny. On/off and beyond - a boolean model of apoptosis. *PLoS Computational Biology* 5 (12) (2009)

27. Steven Watterson, Stephen Marshall, Peter Ghazal. Logic models of pathway biology. *Drug Discovery Today* 13 (9-10), 447–456 (2008)

28. Melody K. Morris, Julio Saez-Rodriguez, Peter K. Sorger, Douglas A. Lauffenburger. Logic-based models for the analysis of cell signaling networks. *Biochemistry* 49 (15), 3216–3224 (2010)

29. Julio Saez-Rodriguez, Luca Simeoni, Jonathan Lindquist, Rebecca Hemenway, Ursula Bommhardt, Boerge Arndt, Utz-Uwe Haus, Robert Weismantel, Ernst D. Gilles, Steffen Klamt, Burkhard Schraven. A logical model provides insights into T cell receptor signaling. *PLoS Computational Biology* (2007)

30. W. V. Quine. The problem of simplifying truth functions. *American Mathematical Monthly* 59 (8), 521–531 (1952)

31. H. Hara, T. Wada, C. Bakal, I. Kozieradzki, S. Suzuki, N. Suzuki, M. Nghiem, E. K. Griffiths, C. Krawczyk, B. Bauer, F. D’Acquisto, S. Ghosh, W. C. Yeh, G. Baier, R. Rottapel, J. M. Penninger. The maguk family protein card11 is essential for lymphocyte activation. *Immunity* 18 (6), 763–775 (2003)

32. J. Ruland, G. S. Duncan, A. Wakeham, T.W. Mak. Differential requirement for malt1 in t and b cell antigen receptor signaling. *Immunity* 19 (5), 749–758 (2003)

33. Tilo Beyer, Mandy Busse, Kroum Hristov, Slavyana Gurbiel, Michal Smida, Utz-Uwe Haus, Kathrin Ballerstein, Frank Pfeuffer, Robert Weismantel, Jonathan A. Lindquist, Burkhard Schraven. Integrating signals from

Discrete, qualitative models of interaction networks

- the t-cell receptor and the interleukin-2 receptor. *PLoS Computational Biology* 7 (8) (2011)
34. Martin J. Brown, Ruchika Nijhara, John A. Hallam, Michelle Gignac, Kenneth M. Yamada, Stanley L. Erlandsen, Jérôme Delon, Michael Kruhlik, Stephen Shaw. Chemokine stimulation of human peripheral blood T lymphocytes induces rapid dephosphorylation of ERM proteins, which facilitates loss of microvilli and polarization. *Blood* 102 (12), 3890–3899 (2003)
35. Raymond Reiter. On closed world data bases. In H. Gallaire, J. Minker, editors, *Logic and Databases*, 55–76. Plenum Press, New York (1978)
36. Keith L. Clark. Negation as failure. In H. Gallaire, J. Minker, editors, *Logic and Databases*, 293–322. Plenum Press, New York (1978)
37. Steffen Klamt, Utz-Uwe Haus, Fabian Theis. Hypergraphs and cellular networks. *PLoS Computational Biology* 5 (5), e1000385 (2009)
38. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Cli_ord Stein. *Introduction to Algorithms*. MIT Press, Massachusetts, 2nd edition (2001)
39. Robert Endre Tarjan. Enumeration of the elementary circuits of a directed graph. *SIAM Journal on Computing* 2 (3), 211–216 (1973)
40. Marco Montalva, Julio Aracena, Anahí Gajardo. On the complexity of feedback set problems in signed digraphs. *Electronic Notes in Discrete Mathematics* 30, 249–254 (2008)
41. Neil Robertson, Paul D. Seymour, Robin Thomas. Permanents, pfa_an orientations, and even directed circuits. *Annals of Mathematics* 150 (3), 929–975 (1999)
42. Victor Klee, Richard Ladner, Rachel Manber. Signsolvability revisited. *Linear Algebra and its Applications* 59, 131–157 (1984)
43. P. P. Dangle, B. Zaharieva, H. Jia, K.S. Pohar. Ras-MAPK pathway as a therapeutic target in cancer—emphasis on bladder cancer. *Recent Pat Anticancer Drug Discov* 4 (2), 125–36 (2009)
44. Kathrin Niermann. *Qualitative Modeling of Signaling Cascades in Cellular Networks*. Diplomarbeit, Otto-von-Guericke Universität Magdeburg, Germany (2007)
45. Klaus Truemper. *Design of Logic-based Intelligent Systems*. John Wiley & Sons (2004)
46. Michael R. Garey, David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco (1979)
47. Philippe Roussel. Prolog, *Manuel de référence et d'utilisation*. Groupe Intelligence Artificielle, Aix-Marseille, France (1975)
48. Utz-Uwe Haus, Kathrin Niermann, Klaus Truemper, Robert Weismantel. Logic integer programming models for signaling networks. *Journal of Computational Biology* 16 (5), 725–743 (2009)
49. Utz-Uwe Haus, Klaus Truemper, Robert Weismantel. Linear satisfiability algorithm for 3CNF formulas of certain signaling networks. *Journal on Satisfiability, Boolean Modeling and Computations (JSAT)* 6, 13–22 (2008)
50. Dimitris Bertsimas, Robert Weismantel. *Optimization over Integers*. Dynamic Ideas, Belmont, Massachusetts, USA (2005)
51. George B. Dantzig. *Linear Programming and Extensions*. Princeton University Press (1963)
52. John N. Hooker. *Integrated Methods for Optimization*. Springer Science+Business Media, LLC, New York, NY (2007)
53. ILOG. CPLEX (1997-2007). <http://www.ilog.com/products/cplex/URL>
<http://www.ilog.com/products/cplex/>
54. Michael L. Fredman, Leonid Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *J. Algorithms* 21 (3), 618–628 (1996)
55. Steffen Klamt, Axel von Kamp. CellNetAnalyzer software package (formerly fluxanalyzer). Available on request. Information at: <http://www.mpi-magdeburg.mpg.de/projects/cna/cna.html>
56. René Thomas. Regulatory networks seen as asynchronous automata: A logical description. *Journal of theoretical Biology* 153 (1), 1–23 (1991)
57. Houssine El Snoussi, René Thomas. Logical identification of all steady states: The concept of feedback loop characteristic states. *Bulletin of Mathematical Biology* 55 (5), 973–991 (1993)
58. Jean-Paul Comet, Hanna Kludel, Stéphane Liauzu. Modeling multi-valued genetic regulatory networks using high-level petri nets. In G. Ciardo, P. Darondeau, editors, *Proceedings of the Int. Conf. on the Application and Theory of Petri Nets*, Lecture Notes in Computer Science, 208–227. Springer-Verlag (2005)
59. Gilles Didier, Elisabeth Remy, Claudine Chaouiya. Mapping multivalued onto boolean dynamics. *Journal of Theoretical Biology* 270 (1), 177–184 (2011)
60. Houssine El Snoussi. Qualitative dynamics of piecewise-linear differential equations: a discrete mapping approach. *Dynamics and Stability of Systems* 4 (3-4), 565–583 (1989)
61. Jesus De Loera, Jon Lee, Peter Malkin, S. Margulies. Hilbert’s Nullstellensatz and an algorithm for proving combinatorial infeasibility. In *Proceedings of ISSAC* (2008)

Discrete, qualitative models of interaction networks

62. David A. Cox, John B. Little, Donald O'Shea. *Ideals, Varieties and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer-Verlag, Berlin, Germany (1992)

63. Niklas Eén, Niklas Sörensson. MiniSat: An extensible SAT-solver. SAT 2003,2005,2007; available at <http://minisat.se/>

64. Utz-Uwe Haus. cl-jointgen, a Common Lisp Implementation of the Joint-Generation Method (2008). Available at <http://www.primaldual.de/cl-jointgen>

65. boost C++ libraries (1999-2011). Available at <http://www.boost.org/>

Key Words: Qualitative modeling, Integer programming, Satisfiability, Monotone boolean functions, Review

Send correspondence to: Utz-Uwe Haus, Institute of Operations Research, Department of Mathematics, ETH Zurich, Ramistrasse 101, CH-8092 Zurich, Switzerland, Tel: 41-44-63-39355, Fax: 41-44-63-21025, E-mail: utz-uwe.haus@math.ethz.ch