Chapter 4

Signaling Networks: Asynchronous Boolean Models

Réka Albert[1] **and Raina Robeva**[2]

[1] *Pennsylvania State University, University Park, PA, USA,* [2] *Sweet Briar College, Sweet Briar, VA, USA*

4.1 INTRODUCTION TO SIGNALING NETWORKS

Living cells receive various external stimuli and convert them into intracellular responses. This process is collectively known as *signal transduction*, and involves a collection of interacting (macro)molecules such as enzymes, proteins, and second messengers [1]. Signal transduction is an important part of a cell's communication with its surroundings. Signal transduction is crucial to the maintenance of cellular homeostasis and for cell behavior (growth, survival, apoptosis, movement). Many disease processes such as developmental disorders, diabetes, vascular diseases, autoimmunity, and cancer [2, 3] arise from mutations or alterations in the expression of signal transduction pathway components.

Figure 4.1 illustrates the characteristic steps of signal transduction. Signal transduction processes are activated by extracellular signaling molecules that bind to receptor proteins located in the cell membrane. The signals are transferred inside the cell through changes in the shape of the receptor proteins and trigger a sequence of biochemical reactions leading to the production of small molecules called second messengers. The signals are amplified through additional biochemical reactions or protein-protein interactions in the cytoplasm, for example phosphorylation of a protein by another protein called a kinase. The information can be passed to the nucleus and can lead to changes in the expression of certain genes. Other signal transduction processes lead to a cellular response at the protein level, such as opening of ion channels. At every step of the signal transduction process feedbacks are possible and are often important.

Many signal transduction processes involve numerous and diverse components and interactions. For this reason it is beneficial to represent them with a network, or graph. The components (e.g., biomolecules) are represented by *nodes* (also called *vertices*), whereas the interactions and processes among the nodes are denoted by *edges* (also called *links*). Edges in the network can be directed, indicating the orientation of mass transfer or of information propagation, and can also have a positive or negative sign to represent activation or inhibition. The totality of the nodes and edges of a network form the *network topology*. This network representation, called a *signal transduction network* or *signaling network*, provides a basis for structural analysis and dynamic modeling of the underlying signal transduction process. These mathematical analyses enable us to trace the propagation of information in the network, to determine the key mediators, and to determine the system's responses under normal circumstances and in the case of perturbations.

Figure 4.2 depicts an example of a real signal transduction network, involved in activation-induced cell death of white blood cells called cytotoxic T cells [5],[6]. This network is of interest because the process of

Algebraic and Discrete Mathematical Methods for Modern Biology. http://dx.doi.org/10.1016/B978-0-12-801213-0.00004-6

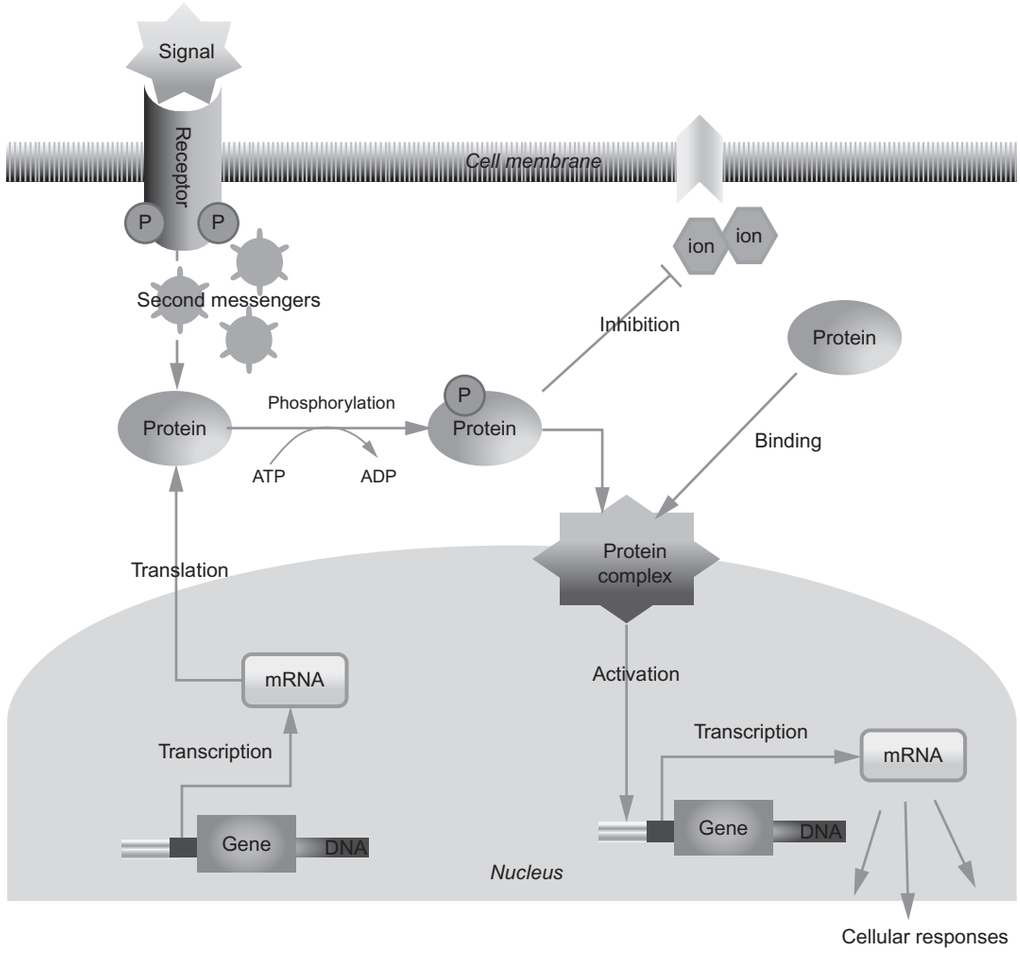

FIGURE 4.1 Scheme of a hypothetical signal transduction process involving diverse interactions of cellular components. Figure reproduced from Ref. [4].

activation-induced cell death is disrupted in the disease T-LGL leukemia, causing the survival of a fraction of activated T cells, which later start attacking healthy cells. In Figure 4.2, the shape of the nodes indicates their cellular location: rectangles indicate intracellular components, ellipses indicate extracellular components, and diamonds indicate receptors. In addition, as this network represents only a small part of the cellular signaling processes, hexagonal nodes are used to summarize its connections with other signal transduction mechanisms or cell behaviors. Such nodes, called conceptual nodes, encapsulate behaviors that are relevant to the network functions. For more details about this specific example see the legend of Figure 4.2.

4.2 A BRIEF SUMMARY OF GRAPH-THEORETIC ANALYSIS OF SIGNALING NETWORKS

A network representation of a signaling mechanism contains essential information, which can be incorporated into its initial analysis. This analysis includes the use of graph-theoretic measures, such as *centrality measures*, *network motifs*, and *shortest paths*, to describe the organization of the network [7].

In signal transduction networks, like in all directed networks, we can categorize the nodes by their incoming and outgoing edges. The nodes with only outgoing edges are called *sources*, and nodes with only incoming edges are *sinks* of the network. Source nodes generally correspond to the signals, while sink nodes denote the outcomes

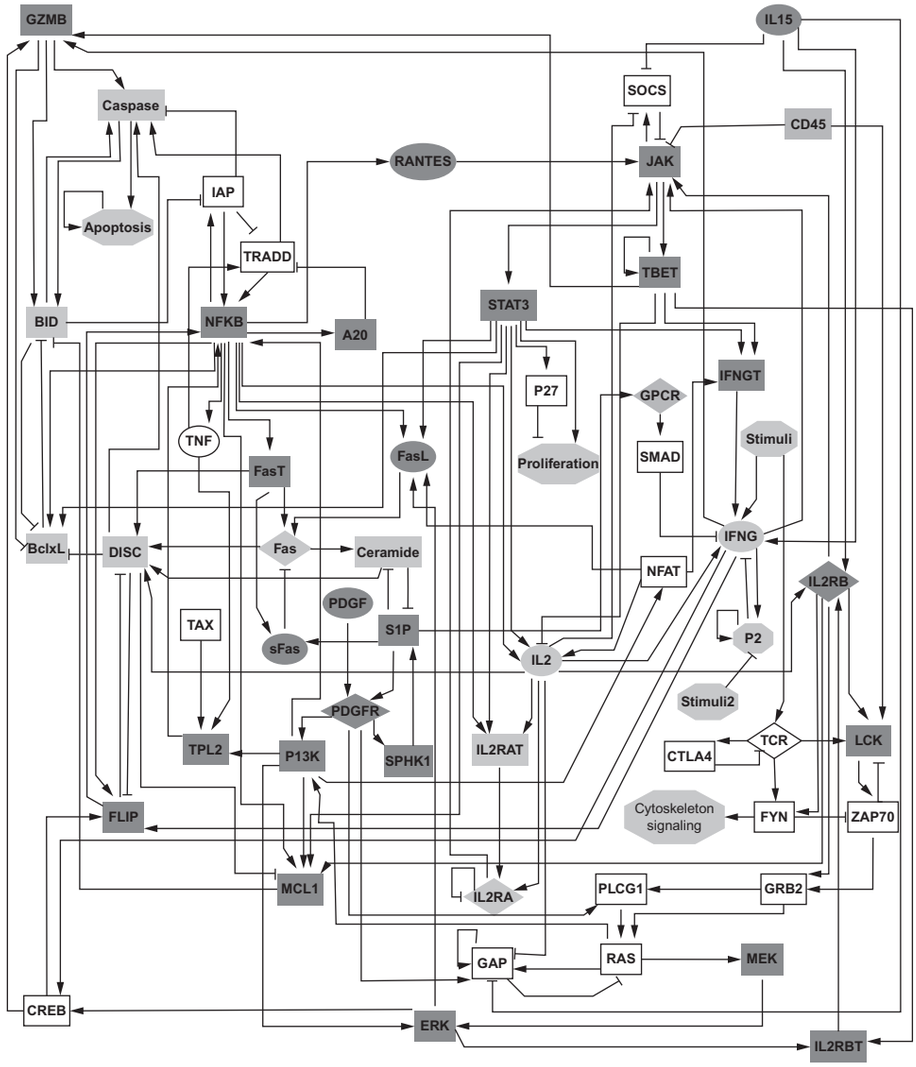

FIGURE 4.2 A signal transduction network involved in activation-induced cell death of white blood cells called cytotoxic T cells. The key signals are Stimuli (representing stimulus of the cell by the presence of pathogens) together with the external molecules interleukin 15 (IL15) and platelet-derived growth factor (PDGF).These signals are identified as nodes with only outgoing links (nodes with this property are called sources). The key output node of the network is Apoptosis, expressing programmed cell death. Note that this node has no outgoing links (a node with this property is called a sink). The nodes of the network include proteins, mRNAs, and concepts. The shape of the nodes indicates the cellular location: rectangles indicate intracellular components, ellipses indicate extracellular components, and diamonds indicate receptors. Conceptual nodes are represented by hexagons. The background of the non-conceptual nodes corresponds to the known status of these nodes in abnormally surviving T-LGL cells as compared to normal T cells: red (dark) indicates abnormally high expression or activity, green (lighter) means abnormally low expression or activity. The full names of the nodes can be found in [5, 6]. An arrowhead or a short perpendicular bar at the end of an edge indicates activation or inhibition, respectively. Figure reproduced from Ref. [6].

of signal transduction networks. In signaling networks it is possible that nodes have an auto-regulatory *loop*, an edge that both starts and ends at the node. Often it is beneficial to extend the definition of source and sink nodes to allow for the presence of a loop. For example, in Figure 4.2 the nodes Stimuli, IL15, and PDGF represent external signals acting on T cells, and indeed they are source nodes. The nodes Apoptosis, Proliferation, and Cytoskeleton signaling represent outcomes of the signal transduction process. Proliferation and Cytoskeleton signaling are sink nodes, and Apoptosis, which has a loop, can also be considered a sink node.

Centrality measures describe the importance of individual nodes in the network. The simplest of such measures is the node *degree*, which quantifies the number of edges connected to each node. For directed networks, the *in-* and *out-degree* of a node is defined as the number of edges coming into or going out of the node, respectively. The nodes whose combined in- and out-degrees are in the top 1% to 5% of all the nodes are termed *hubs*. These hub nodes often play an important role in the network. For example, the node representing the NFκB protein is a hub of the T-LGL network on Figure 4.2, having an out-degree of 11 and an in-degree of 5. This is not surprising, as NfκB is a transcription factor that is known to be important in cellular responses to various stimuli and in cell survival (see, e.g., Ref. [8]).

From a graph-theoretic standpoint, a *path* is a sequence of adjacent edges in the network. In networks that can have both positive and negative edges, the *sign of a path* is positive if there are no or an even number of negative edges in the path and is negative if there is an odd number of negative edges. A path containing two or more edges that begins and ends at the same node is called a *cycle*. The *length* of a path or a cycle is defined to be the number of its edges (loops can be considered as cycles of length one).

Network motifs are recurring patterns of interconnection with well-defined topologies [9]. Among these motifs are *feed-forward loops* (in which a pair of nodes is connected by both an edge or short path and a longer path) and *feedback loops* (directed cycles). An example of a feed-forward loop in Figure 4.2 is the subgraph formed by the nodes STAT3, P27, and Proliferation; this is an incoherent feed-forward loop, as the STAT3–Proliferation edge is positive and the path between them is negative. An example of a positive feedback loop on Figure 4.2 is the directed cycle between S1P, PDGFR, and SPHK1, while the cycle between TCR and CTLA4 is a negative feedback loop. Feed-forward loops are more abundant in the transcriptional regulatory and signaling networks of different organisms compared to randomized networks that keep each node's degree. They have been found to support several functions, such as filtering of noisy input signals, pulse generation, and response acceleration [9]. Positive feedback loops were found to support multistability while negative feedback loops could cause pulse generation or oscillations [10]. Examples that illustrate such behaviors will be presented in Section 4.7.

A signaling network, as all directed networks, is *strongly connected* if, for any two nodes in the network u and v, there is a directed path from u to v and another path from v to u. If a network is not strongly connected, it is informative to identify *strongly connected components* (or subgraphs) of the network. Having no strongly connected components (SCCs) indicates that the network has an acyclic structure (i.e., it does not contain feedback loops), while having a large SCC implies that the network has a central core. Signaling networks tend to have a strongly connected core of considerable size [11]. For example, the network in Figure 4.2 has a strongly connected component of 44 nodes, which represents 75% of all nodes. An SCC may have an in-component (nodes that can reach the SCC) and out-component (nodes that can be reached from the SCC). In biology, nodes in each of these subsets tend to have a common task. In signaling networks, the nodes of the in-component represent signals or their receptors and the nodes of the out-component are usually responsible for the transcription of target genes or for phenotypic changes [11]. The out-component of the T-LGL network in Figure 4.2 consists mainly of conceptual nodes that represent cell behaviors, such as apoptosis (the genetically determined process of cell destruction) or proliferation (the increase in the number of cells due to cell growth and division).

Software packages for network visualization and analysis include yEd Graph Editor, available from http://www.yworks.com/en/products/yfiles/yed/. Cytoscape[12], NetworkX [13], and Pajek [14].

Exercise 4.1. Consider the network depicted in Figure 4.3.

1. Is the network strongly connected? Explain your answer.
2. If the network is not strongly connected, identify its strongly connected components.
3. Does the network contain loops? If so, identify them.
4. Does the network contain cycles? If so, identify all cycles.
5. Are there any feed-forward loops? If so, identify them as positive, negative, or incoherent.
6. Are there any feedback loops? Identify them. Identify their sign as positive or negative. □

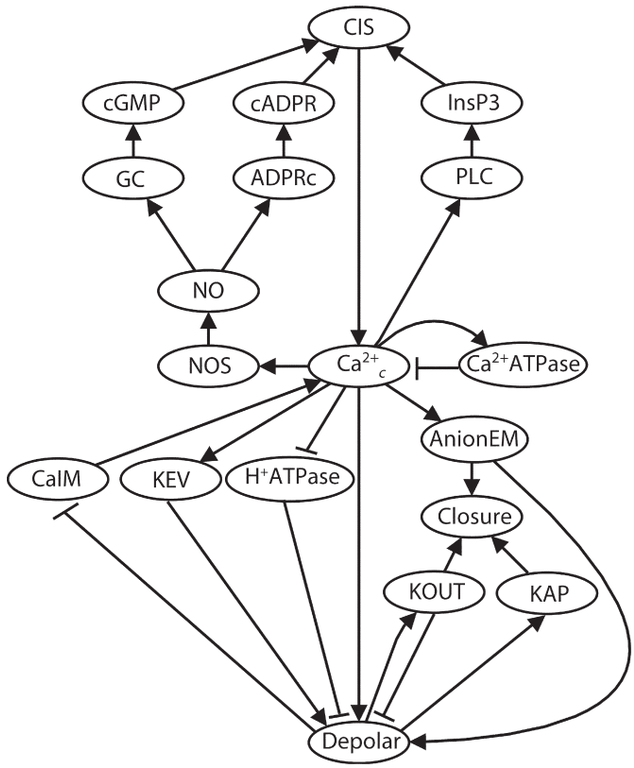

FIGURE 4.3 Figure for Exercise 4.1. The network is a part of a plant signal transduction network whose signal is the drought hormone abscisic acid and whose outcome is the closure of the stomata (microscopic pores on the leaves) [15, 16]. An arrowhead or a short perpendicular bar at the end of an edge indicates activation or inhibition, respectively. The full names of the nodes can be found in [15, 16]. Figure reprinted from Ref. [17] with permission from Elsevier.

4.3 DYNAMIC MODELING OF SIGNALING NETWORKS

Representation as a network of nodes connected pairwise by edges offers a coherent representation of a system of interacting biomolecules [7, 17, 18]. Going further, a *dynamic model* can describe how the abundances of the biomolecules in the network change over time due to their interactions. This is done by associating each node i of the signaling network with a variable x_i. Dynamic modeling approaches can be continuous or discrete according to the use of continuous or discrete variables. Continuous dynamic modeling [18, 19] describes the rate of change of each continuous variable x_i as a function of the other variables x_j in the signaling network. These models require the knowledge of mechanistic details for each interaction (e.g., the stoichiometric coefficients of the molecules that participate in a reaction and the kinetic rate functions) and their parameterization with rate constants. One also needs quantitative measurements of all the variables in the system in the initial condition and also in at least one stable state, to use for model validation. Continuous modeling is most feasible for well-characterized systems, where through decades of experimental work a sufficient amount of quantitative information has been gathered. Unfortunately, the state of the art for most systems is far from this mark: in many cases not all interactions have been mapped out, the detailed mechanisms are not known, there are no quantitative measurements of all the relevant variables, and kinetic parameter values are unknown and difficult to estimate. Continuous modeling is not practical for these types of systems.

As an alternative, discrete dynamic modeling such as Boolean network models [10, 20], multivalued logical models [21], and Petri nets [22] have been developed. These models use discrete variables that correspond to logic categories rather than quantitative values and describe the future value of a variable x_i (as opposed to

its rate of change) as a function of the other variables x_j in the signaling network. Discrete dynamic models can be constructed from qualitative or relative measurements (e.g., whether a protein is more active in one condition compared to another), have no or very few kinetic parameters, and are able to provide a qualitative dynamic description of the system. They can be used to elucidate how perturbations may alter normal behavior and thus lead to testable predictions which are especially valuable in poorly understood large-scale systems. These approaches can be employed for systems with hundreds of components and have been used to model signal transduction networks in unicellular organisms, plants, animals, and humans (reviewed in Ref. [4]).

This chapter focuses on Boolean models. Construction of a Boolean model starts with a compilation of a list of components (nodes) and of the known interactions and regulatory relationships among these nodes, which will become the edges of the reconstructed interaction network. The model construction continues with determining Boolean functions which describe the regulation of each node based on the edges incident on the node and also using information from the literature. The collection of the resting or pre-stimulus states of the nodes will be used as an initial condition in the model. The model construction also includes a choice of how to represent the passing of time; as we'll see below, this choice has a subtle influence on certain outcomes of the model. Having chosen the transition functions, initial conditions, and the representation of time, running the model will provide a simulation of how the system evolves in time.

The model-indicated dynamic behaviors resulting from the simulations (e.g., long-term states) need to be compared with the available biological information on the behaviors of the system. If there are qualitative discrepancies that cast doubt on the model, the edges or Boolean functions of the model need to be rechecked and suitably revised. On the other hand, qualitative agreement between the model's results and biological knowledge increases our confidence in the model and allows its use to generate understanding and new predictions. For example, an often-used follow-up is a comprehensive analysis of the effects of node perturbations. We next describe the modeling process in detail.

4.4 THE REPRESENTATION OF NODE REGULATION IN BOOLEAN MODELS

The Boolean model of a signaling network associates each network node (i.e., gene, protein, molecule) i with a binary variable x_i which describes its expression level, concentration, or activity. The value $x_i = 1$ (ON) represents that component i is active or expressed, or has an above-threshold concentration; the value $x_i = 0$ (OFF) denotes that it is inactive or not expressed, or has a below-threshold concentration. The thresholds invoked in the definition of states do not need to be quantified, as long as it is known that a concentration level exists above which the component in question can effectively regulate its downstream targets. In Boolean models, the future state of node i, denoted by x_i^*, is determined based on a logic statement involving the current states of its regulators, i.e., $x_i^* = f_i$. This statement f_i, called a *Boolean transition function* (or a *Boolean rule*), represents the conditional dependency of the input (regulator) nodes in the regulation of the downstream (target) node. This function is usually expressed via the logic operators AND, OR, and NOT. For example, $f_4 = (x_1 \text{ OR } x_2) \text{ AND } (\text{NOT } x_3)$ is a Boolean function regulating the variable x_4. It indicates that x_4 will be ON when at least one of x_1 or x_2 is ON and simultaneously x_3 is OFF. When parentheses are used, as in this example, they determine the order of operations explicitly. Alternatively, the order of precedence of the logical operators may be used: NOT has the highest precedence, followed by AND, and then by OR, which has the lowest precedence. Any use of parentheses overrides the precedence rules. As an example, the rule for f_4 above can also be written as $f_4 = (x_1 \text{ OR } x_2) \text{ AND NOT } x_3$.

A Boolean transition function can also be represented by a truth table, in which each row lists a possible combination of state values for the node's regulators and the associated output value of the function. Table 4.1 presents the truth tables for the Boolean functions corresponding to the operations NOT (third column), OR (fourth column), and AND (last column). The truth table of a Boolean function with k variables has 2^k rows and $k + 1$ columns (see Exercise 4.3).

Example 4.1. Consider the three-node signal transduction network depicted on Figure 4.4a. The source node A is the signal to the network, and both A and B positively regulate the sink node C. Let's identify transition functions compatible with this network.

TABLE 4.1 Truth Tables Illustrating the NOT, OR, and AND Operators

x_A	x_B	f_C = NOT x_A	$f_D = x_A$ OR x_B	$f_E = x_A$ AND x_B
0	0	1	0	0
0	1	1	1	0
1	0	0	1	0
1	1	0	1	1

The third column (f_C) indicates that the value of "NOT x_A" is the opposite (logic negation) of the value of x_A. The fourth column (f_D) indicates that "x_A OR x_B" is 1 whenever either input is 1. The last column (f_E) indicates that "x_A AND x_B" is 1 only when both inputs are 1.

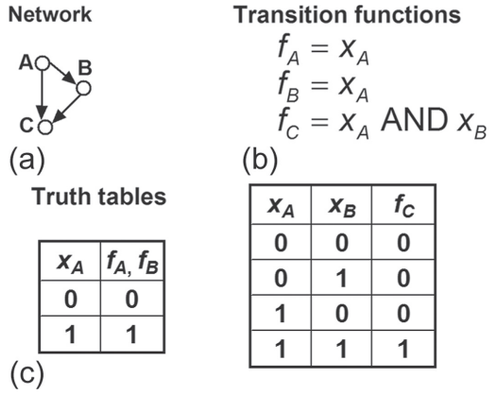

FIGURE 4.4 A Boolean model of a simple signal transduction network. (a) The signal transduction network. The edges with arrows represent positive effects. Note that the network does not uniquely determine the Boolean updating function for node C. (b) The Boolean transition functions in the model. The first transition function indicates that the state variable of node A does not change. The second transition function indicates that the state variable of node B follows the state of node A with a delay. The third transition function indicates that the condition for the ON state of node C is that both A and B are on. (c) The truth tables of the Boolean updating functions given in (b).

Because node A is the signal to the network, we are free to choose its transition function as long as it is independent of the other nodes. Let us assume that the state of node A stays constant, maintaining whatever value it started from. The corresponding transition function is $f_A = x_A$, and the equation governing the state of node A is $x_A^* = f_A = x_A$. Node B is positively regulated by node A, and is not regulated by anything else, thus its transition function is $f_B = x_A$. This indicates that the state of node B will follow the state of node A with a delay, $x_B^* = f_B = x_A$. Node C is positively regulated by node A and by node B. The network does not tell us how these two influences are cumulated. There are two choices: $f_C = x_A$ OR x_B, and $f_C = x_A$ AND x_B. The first function indicates that either A or B alone can successfully activate C (see column four of Table 4.1), while the second expresses the more stringent condition that both A and B need to be on simultaneously in order to activate C (see the last column of Table 4.1). Figure 4.4 indicates one of the two possible sets of transition functions, both as Boolean expressions (b) and as truth tables (c). □

Exercise 4.2. Construct the transition function and truth table for the networks in Figure 4.5. Consider both the AND and OR possibilities for the transition function of C for the network in Figure 4.5a and for the transition function of the node B for the network in Figure 4.5b. □

Exercise 4.3. Show that the truth table of a Boolean function with k variables has 2^k rows and $k + 1$ columns. *Hint*: Determine the number of different sequences of length k that can be formed from 0s and 1s. □

4.5 THE DYNAMICS OF BOOLEAN MODELS

In a Boolean model of a signal transduction network, time is usually discrete. This means that the model variables are updated only at fixed-time instances separated by a certain number of time steps and that no updates are being made between the time steps. As time is implicit in most Boolean models, one could think of a digital clock running in the background, with updates occurring only when the time on the clock changes. The time step between updates could vary from fractions of a second to hours, depending on the nature of the biological system [9]. Mathematically, the state of the system containing n nodes with associated state variables $x_1, x_2, \ldots, x_n$ at time t can be represented by a vector $(x_1(t), x_2(t), \ldots, x_n(t))$ with the ith element representing the state of node i at time t. By successively reevaluating each node's state while applying the corresponding transition function, the system's collective state evolves over time and eventually reaches a *steady state* (i.e., a state that remains unchanged over time) or a set of recurring states. These steady or recurring states are collectively referred to as *attractors*. Attractors that are not steady states are called *complex attractors*. For each attractor, its *basin of attraction* is comprised of all states that eventually lead to the attractor.

Exercise 4.4. Can you guess the attractor(s) of the Boolean model in Example 4.1? Consider the cases $x_A = 0$ and $x_A = 1$ separately. □

The transition functions of a Boolean model specify the rules for updating the network variables, but the order in which the updates are performed needs to be indicated separately. Various update schedules can be implemented via synchronous or asynchronous update algorithms. The *synchronous scheme* is the simplest update mode, wherein the states of all nodes are updated simultaneously according to the state of the system at the previous time step [20]. One significant disadvantage of this type of update is that it implicitly assumes that the timescales of all biological events in the system are similar and that the state transitions of components are synchronized. However, many systems include a mixture of biological events of different timescales (e.g., from fractions of seconds for protein-protein interactions to several minutes for transcription [9]), making the use of synchronous update inappropriate in those systems.

Asynchronous models aim to account for timescale diversity by updating the nodes in a nonsynchronous manner. There are *deterministic asynchronous* schemes with fixed individual timescales or fixed time delays. There also are *stochastic asynchronous* schemes wherein each node is updated with a certain probability, all nodes are updated according to a random sequence, or one randomly selected node is updated at a time step [23]. A parsimonious way to deal with diverse and unknown timescales is to use stochastic asynchronous update and do many simulations. We next present several examples that illustrate the different update algorithms.

Example 4.2. Assume that we have a network composed of nodes A and B. The edges do not matter in this example. Let's construct two deterministic and two stochastic updating schemes for this network.

In synchronous update, both nodes will be updated simultaneously at multiples of a time step, i.e., at time instances 1, 2, 3, …, t. If we are currently at time step t, the future state of a node means the state at time $t + 1$. Thus the state transitions of the two nodes will be $x_A^* = x_A(t + 1) = f_A(t)$, $x_B^* = x_B(t + 1) = f_B(t)$.

For another example of deterministic update, let us assume that node A can change state at multiples of a timescale t_A, while node B can change state at multiples of a timescale t_B. For simplicity let's designate the

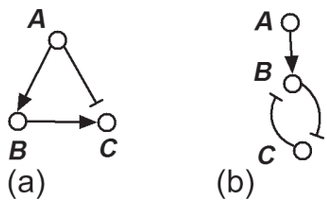

FIGURE 4.5 Figure for Exercise 4.2 and several of the follow-up exercises. Two simple signal transduction networks. For both networks, A is the source node (signal).

smaller timescale as the unit, and express the bigger timescale as an integer multiple of the smaller, for example, $t_A = 1$, $t_B = 2$. This means that node A will be updated at every time step while node B will be updated at even time steps. So the update scheme is A, A and B together, A, A and B, A, A and B….

A popular stochastic update, called *random order asynchronous update*, is performed as follows: (1) generate a permutation of the nodes and update them once in this order; (2) generate another permutation and update the nodes in the new order; (3) continue, by selecting a random permutation of the nodes at each step and updating the nodes in the order indicated by the permutation. The order of update in our example may look something like this: A, B; A, B; B, A; A, B; …where the semicolons indicate the end of a time step, which here is interpreted as a round of update. Whenever node B is updated, it uses the state of node A obtained at its most recent update, which can be within the same round of update (such as in the first two rounds of our example). The same is true for node A. Notice the difference here in comparison with the synchronous update schedule where the update of B at time $t + 1$ always uses the value of A at time t.

Another frequently used stochastic update method is to update one randomly selected node at each time step. This method is called *general asynchronous update*. The order of update of our two nodes may look like this: B; A; A; A; B; B; …When considering a long sequence of updates, it is possible that node A is updated more (or less) than node B using this method, while under random order asynchronous update they will be updated the same number of times. Nevertheless, because the probability of choosing each node is equal, on average they will be updated the same number of times. If we know that one node should be updated more frequently, we can choose unequal selection probabilities. □

A compact representation of all possible trajectories is visualized through the *state transition graph*, whose nodes are states of the system and whose edges denote the allowed transitions among the states according to the chosen updating scheme [23]. The attractors of the system can be determined from network analysis of the state transition graph. Fixed points will correspond to states that do not have any outgoing edges (transitions), only a loop. Each complex attractor forms a terminal strongly connected component of the state transition graph (i.e., a strongly connected component with an empty out-component).

Example 4.1. (continued). Consider again the signaling network and Boolean model in Figure 4.4. Let's represent the system's state as the triple $x_A\ x_B\ x_C$. Let us first determine the system's state transition graph in the absence of a signal ($x_A = 0$) when using synchronous update. We start from an initial state, let's say 011, and update each node's state using their corresponding transition functions. It is easiest to look up the function's output from the truth tables. Node A will keep its OFF ($x_A = 0$) state. The next state of node B is indicated by the first row of the truth table on the left, giving 0. The next state of node C can be looked up from the second row of the truth table on the right, yielding 0. Thus the next state of the system is 000. We have so far obtained the first edge of the state transition graph, from state 011 to state 000. Let's start from 000. Checking the first row of both truth tables, we find that the state remains 000. This state is thus a steady state (fixed point) of the system. Did you guess this state in Exercise 4.4?

We still have two states to consider as initial conditions. Let's start from 010. From the first row of the left truth table, the next state of node B is 0, and from the second row of the truth table on the right, the next state of node C is also 0. So the state 010 transitions to 000. Finally, starting from state 001, we find 000 as well. All initial conditions in which node A is OFF transition to the 000 steady state, thus the state transition graph has four nodes and four edges, starting from each state and all ending in 000, as shown in the left panel of Figure 4.6a. □

Exercise 4.5. Determine the state transition graph of the model in Figure 4.4 in the presence of a signal ($x_A = 1$) when using synchronous update. Compare with Figure 4.6a (right panel). □

Exercise 4.6. Determine the state transition graphs for the networks in Figure 4.5, assuming synchronous update. Consider both the AND and OR possibilities for the transition function of C for the network in panel (a) and for the transition function of B for the network in panel (b). Consider both the sustained absence ($x_A = 0$) and presence ($x_A = 1$) of node A. □

Example 4.1. (continued). Let us consider again the network from Figure 4.4 with the absence of signal ($x_A = 0$) but use general asynchronous update, when one node is updated at any given time step.

Because node A does not change state, its update does not need to be considered. But either node B or node C can be updated with equal probability, so the state transition graph will need to include both transitions. In general, the maximum number of transitions from any given state equals the number of nodes in the network.

Let's start with state 011 as we did for synchronous update. To update node B, we look up its next state from the first row of the truth table in Figure 4.4 on the left, and find that it is 0. The next state is thus 001, having the same state for node A (which does not change) and for node C (which was not updated). If we start from state 011 and update node C, its next state is 0, thus the next state of the system is 010. Thus state 011 has two successors, namely, 001 and 010, which is a markedly different result than the single successor, 000, which we found when using synchronous update (see Figure 4.6b, left panel, and compare with the state transition graph in Figure 4.6a, left panel). Indeed, the transition under synchronous update involved the state change of both node B and C, which is not possible under general asynchronous update.

The other two transitions we found for synchronous update involve a single state change, thus it is not a surprise to find that they are preserved. (Check this result.) When updating node B in state 001, or updating node C in state 010, the state remains unchanged, thus the transition is represented by a loop (Figure 4.6b, left panel). □

Exercise 4.7. Determine the state transition graph of the model in Figure 4.4 in the presence of signal ($x_A = 1$) when using general asynchronous update. Compare with Figure 4.6b (right panel). □

Figure 4.6 summarizes the state transition graph corresponding to synchronous update (a) and general asynchronous update (b). The asynchronous state transition graph has more edges, because there are up to twice as many distinct transitions when updating two nodes one by one instead of simultaneously. Most of the extra edges are loops and correspond to updates when a node's state is reevaluated but does not change. Nevertheless, the states that do not have any outgoing edges in addition to loops, i.e., the steady states 000 and 111, are identical for both types of update. Is this result generally true? Or do we expect any changes in the system's steady state if we switch from synchronous to asynchronous update? We will explore this question in the next section.

Exercise 4.8. Determine the state transition graphs for the networks in Figure 4.5 when using general asynchronous update. Consider both the AND and OR possibilities for the transition function of C for the

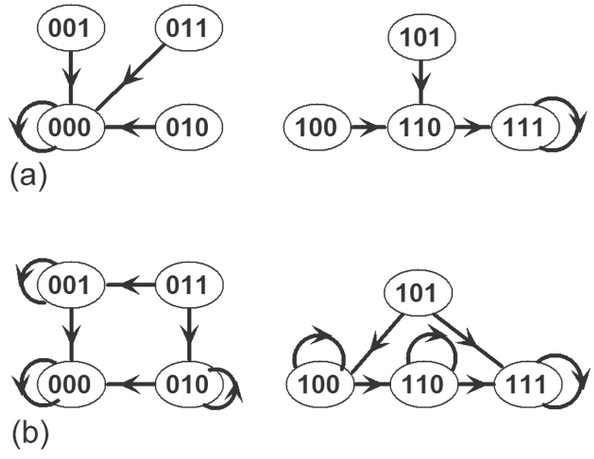

FIGURE 4.6 State transition graphs corresponding to the Boolean model presented in Figure 4.4. The symbols correspond to the states of the system, indicated in the order A, B, C; thus 000 represents $x_A = 0$, $x_B = 0$, $x_C = 0$. A directed edge between two states indicates the possibility of transition from the first state to the second by updating the nodes in the manner specified by the updating scheme. An edge that starts and ends at the same state (a loop) indicates that the state does not change during update. (a) The state transition graph corresponding to synchronous update, when all nodes are updated simultaneously. The two states that have loops are the fixed points of the system. (b) The state transition graph corresponding to updating one node at a time (general asynchronous update). While several states have loops, indicating that at least one of the nodes does not change state during update, only the two states that have no outgoing edges are fixed points of the system.

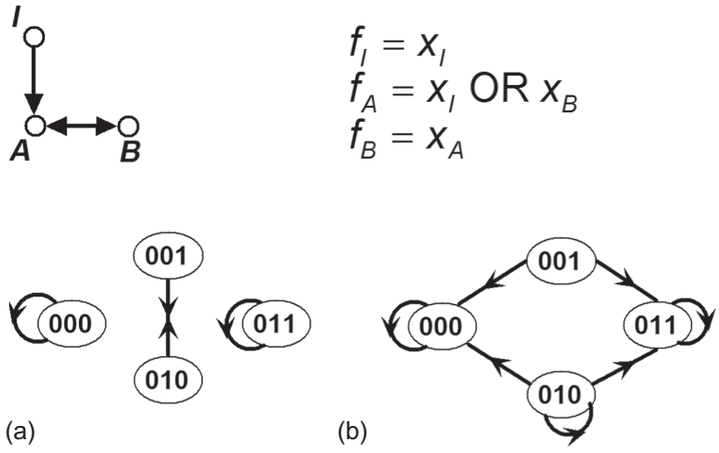

FIGURE 4.7 A simple signal transduction network composed of a source node I and two nodes, A and B, which form a mutual activation loop. It is assumed that the positive inputs from I and B are independently sufficient to activate node A. (a) The network's state transition graph corresponding to synchronous update, when the signal is OFF ($x_I = 0$). The states are specified in the node order I, A, B. (b) The state transition graph corresponding to general asynchronous update, when the signal is OFF ($x_I = 0$).

network in Figure 4.5a and for the transition function of B for the network in Figure 4.5b. Consider both the sustained absence ($x_A = 0$) and presence ($x_A = 1$) of node A. □

Exercise 4.9. For each of the cases considered in Exercises 4.6 and 4.8, compare the steady states obtained when using synchronous update and general asynchronous update. Are the steady states the same? □

Exercise 4.10. Consider the network in Figure 4.7. Determine the state transition graph of the network, using first synchronous update and then general asynchronous update. For the case $x_I = 0$, compare your graphs with those in Figure 4.7. □

4.6 ATTRACTOR ANALYSIS FOR STOCHASTIC ASYNCHRONOUS UPDATE

As we have already stated, attractors fall into two groups: fixed points (steady states), wherein the state of the system does not change, and complex attractors, wherein the system oscillates, regularly or irregularly, among a set of states. Fixed-point attractors usually correspond to the steady activation states of components or to cellular phenotypes in signaling networks. For example, the three fixed points of a Boolean model of a T-helper (Th) cell differentiation network [24] recapitulated the activation patterns of components observed in Th0, Th1, and Th2 cells, respectively. Complex attractors correspond to cyclic and oscillatory behaviors such as the cell cycle, circadian rhythms, or Ca^{2+} oscillations. The qualitative features of Boolean modeling make it suitable for analyzing the repertoire of behaviors in a large-scale system, such as its possible multistability (the existence of multiple stable steady states) [25], the initial conditions that lead to one attractor versus the other, and the activity changes of components following a perturbation. For example, the Boolean model of the T-cell apoptosis signaling network shown in Figure 4.2 indicated the existence of two fixed-point attractors, one corresponding to apoptosis and the other to a survival state which embodies the abnormal T-cell fate seen in the disease T-LGL leukemia [5, 6]. Analysis of the model also indicated the minimal perturbation that leads to the emergence of the abnormal survival state. Interestingly, this minimal perturbation involves only the overexpression of two external signals, IL15 and PDGF, suggesting that this disease does not necessarily have a genetic component. In this section we examine in what ways different update schedules may affect the attractors of a Boolean model. We consider fixed points (steady states) first.

Notice that arriving at a steady state in the state transition graph means that (after sufficiently many time steps) all of the system variables become constant with respect to time, i.e., $x_i^* = x_i$ for all nodes i. Thus, as the fixed points of a system are time independent, they are in fact the same for both synchronous and asynchronous

updates. This also means that there is an alternative way to determine them. Because by definition $x_i^* = f_i$, the condition $x_i^* = x_i$ for all i leads to the set of equations $f_i = x_i$ for all nodes i, which we can solve. This is relatively easy if the network is small; for example, one can use elimination of variables. There are also advanced methodologies, such as transforming the Boolean equations into polynomial equations and solving them using Groebner bases (see Chapters 1 and 3 of [26]).

Exercise 4.11. Determine the steady states of Example 4.1 by solving the set of equations $f_A = x_A$, $f_B = x_B$, $f_C = x_C$. □

Example 4.3. Consider the hypothetical signal transduction network and its associated Boolean model in Figure 4.7. Let us compare the state transition graphs corresponding to synchronous update (a) and general asynchronous update (b) when $x_I = 0$. The determination of these state transition graphs was the subject of Exercise 4.10.

The synchronous state transition graph has two steady states (000 and 011) and a cyclic attractor formed by the states 001 and 010. The asynchronous state transition graph has the steady states 000 and 011, and no additional attractors. Indeed, synchronous models may exhibit limit cycles that are not present in stochastic asynchronous models. These limit cycles depend on two or more variables changing state at the same time. This synchronization among variables is not robust to stochasticity. The disappearance of the cyclic attractor also causes a change in the basins of attraction of the two steady states. Even if no attractors are lost when introducing asynchronicity, the choice of updating scheme can affect the attractors' basins. Because each state has a single successor under synchronous update, the attractors of a synchronous model have disjoint basins. But as stochastic asynchronous update allows several successors of a state, it is possible that a state is in the basin of two or more attractors. For example, in Figure 4.7b states 001 and 010 are in the basin of both steady states. □

Exercise 4.12. For each of the networks considered in Exercises 4.6 and 4.8,

1. Compare the complex attractors obtained when using synchronous update and general asynchronous update. Are they the same?
2. Find the basins of attraction for each of the steady states. □

Several software tools are available for Boolean dynamic modeling of biological systems. BooleanNet [27] can be used to simulate synchronous and random order asynchronous models and to determine their state transition graph. The R package BoolNet [28] provides attractor search and robustness analysis methods for synchronous, asynchronous, and probabilistic Boolean models. SimBoolNet, a plugin to the biological network analysis tool Cytoscape [12], determines state trajectories and attractors using sequential update (starting from the external signals). The software ADAM [29] performs analysis of synchronous Boolean models as examples of polynomial dynamical systems. For networks with less than 20 nodes ADAM can generate the full state transition graph. For larger networks it indicates the fixed points and limit cycles of user-specified length.

Exercise 4.13. Consider the network in Figure 4.8.

1. Determine the state transition graph and the attractors of the network using synchronous update. Find the basins of attraction for each attractor.
2. Repeat number 1, now using general asynchronous update of the node states. □

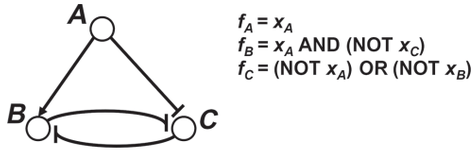

FIGURE 4.8 A simple three-node network for Exercise 4.13.

4.7 BOOLEAN MODELS CAPTURE CHARACTERISTIC DYNAMIC BEHAVIOR

Having now introduced the basic characteristics and properties of Boolean networks, it is important to ask if they can be used as realistic qualitative approximations of signal transduction dynamics in biology. In this section we give examples that demonstrate the ability of Boolean models to capture complex dynamic behavior such as excitation, adaptation, and multistability, which are common in signaling networks. For this section we assume that the reader has basic familiarity with continuous models described by ordinary differential equations. For a detailed introduction to the modeling of biochemical reactions with differential equations see Chapter 2, Section 3, of [26].

As we mentioned in Section 4.2, positive feedback loops support multistability [10], coherent feed-forward loops support the filtering of noisy input signals, and incoherent feed-forward loops support excitation-adaptation behavior [9, 25]. The review article [25] gives examples of continuous dynamic models exhibiting a transient excitation-adaptation behavior based on an incoherent feed-forward loop (Figure 4.9, top row) and a bistable response based on a positive feedback loop (Figure 4.9, bottom row). Let's see if Boolean models based on the same network motifs can qualitatively reproduce these behaviors.

Excitation-adaptation behavior is frequently observed in chemotaxis, which means cells' motion toward a chemical attractant, and can be based on a negative feedback loop [30] or an incoherent feed-forward loop [31].

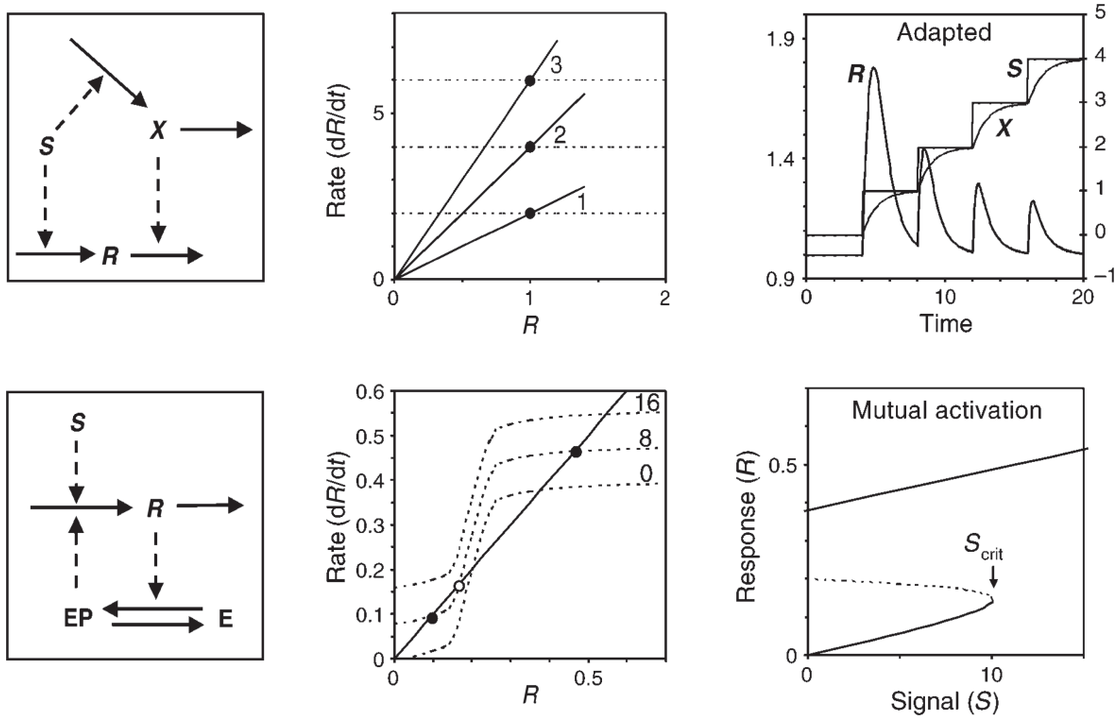

FIGURE 4.9 In the networks (left column), solid edges represent mass flow, such as synthesis or degradation of a protein, and dashed edges represent regulation of synthesis or of degradation. The source node of the networks represents the signal, and one node is designated as the response. The top row represents an example of feed-forward loop-based excitation-adaptation behavior, with *S* denoting the strength of the signal and *X* and *R* denoting the concentrations of two proteins. In the bottom network, the cycle between the *EP* and *E* nodes represents a phosphorylation-dephosphorylation cycle in which the total protein concentration is constant. This cycle is described by Michaelis-Menten kinetics, while the rest of the processes are assumed to follow mass action kinetics. The diagrams in the middle column indicate the absolute value of the rate of the synthesis and degradation of protein *R* as a function of the concentration of *R*. The diagram in the top right indicates the time course of the concentration of protein *R* when the value of the signal is increasing in a step-wise manner. The diagram in the bottom right indicates the steady state concentration of protein *R* as a function of the value of the signal. Figure reprinted from Ref. [26] with permission from Elsevier.

An example of feed-forward loop-based excitation-adaptation behavior is the model shown in the top row of Figure 4.9. In a continuous model, such processes are usually modeled by differential equations describing the dynamics of the concentrations of the biomolecules that make up the system. These models are continuous with respect to both time and the values of the model variables, and the model equations describe the rate of change for each of the variables as a function of all of them. The equations describing the continuous model for our example are

$$\frac{\mathrm{d}R}{\mathrm{d}t} = k_1 S - k_2 X R$$

$$\frac{\mathrm{d}X}{\mathrm{d}t} = k_3 S - k_4 X$$

where S is the value of the signal, X is the concentration of protein X, R is the concentration of the output protein R, and k_i, $i = 1 \ldots 4$ are rate constants. The analysis of this model, discussed in [25], yields that the steady-state (time-independent) concentration of R does not depend on the value of the signal. When the value of the signal undergoes step changes, the concentration of R increases transiently, peaks, then decreases to the signal-independent steady state (see top rightmost panel in Figure 4.9).

Example 4.4. Let's construct a Boolean model of the incoherent feed-forward loop of Figure 4.9 (top left). We start constructing the network of interactions by redefining the edges such that they represent regulatory relationships among the three nodes, S, X, and R. Because S catalyzes the synthesis of R, there is a positive relationship between S and R, indicated by a directed and positive edge starting from S and ending in R (Figure 4.10). Similarly, there needs to be a positive edge starting from S and ending in X. X catalyzes the degradation of R; thus the relationship between X and R is negative, indicated by a negative edge starting from X and ending in R. The uncatalyzed (free) degradation of X and R may each be represented as a negative loop at the respective node, but such degradation is usually left implicit when showing the networks that underlie Boolean models. The reason for this is that decay is implicitly incorporated as a default in all transition rules that depend only on the node's regulators. For example, $f_R = x_S$ already implements that x_R decays to OFF if x_S is OFF, even if x_R was ON before. The absence of decay, on the other hand, should be incorporated explicitly by including the node's own state variable in the transition rule, $f_R = x_S$ OR x_R. This case would be shown in the network by adding a positive loop to R.

Figure 4.10 illustrates the network underlying the Boolean model. We can assume that x_S stays constant ($f_S = x_S$), the Boolean rule of X is unambiguous ($f_X = x_S$), and the transition function of R closest to the continuous model is $f_R = x_S$ AND NOT x_X. The latter holds because S and X affect R simultaneously (that is, the concentration of R is the result of two simultaneously ongoing processes: the increase of the concentration due to the stimulus and the decrease of the concentration due to the catalyzed degradation of R by X). Let's assume that X and R have similar timescales, and use synchronous update. □

Exercise 4.14. Determine the state transition graph for Example 4.4 when using synchronous update. Compare with Figure 4.10b. Does the steady state of R depend on the state of the signal S? □

Now let's reproduce a step-wise increase in the signal variable x_S. The only such choice in a Boolean model is from $x_S = 0$ to $x_S = 1$. Starting with $x_S = 0$ and an arbitrary state for X and R, the system goes into the steady state 000 in one step. Let's now set $x_S = 1$, leading to the state 100 (see transition shown with dashed lines in Figure 4.10b). The state 100 is not an attractor, so the system's state will change into 111, wherein $R = 1$. Thus the step change in x_S drove x_R to change from 0 to 1 (excitation for R). The next state is 110, which is a steady state of the system. In this steady state, x_R is 0 (adaptation for R). Thus the Boolean model qualitatively reproduces the excitation-adaptation behavior: the change in x_S drove a transient excitation of x_R, but the steady state value of x_R was the same for both values of x_S. □

Exercise 4.15. Consider the steady state 110 in Figure 4.10b and implement a step change in S from 1 to 0. Is there an excitation (state change) in R? □

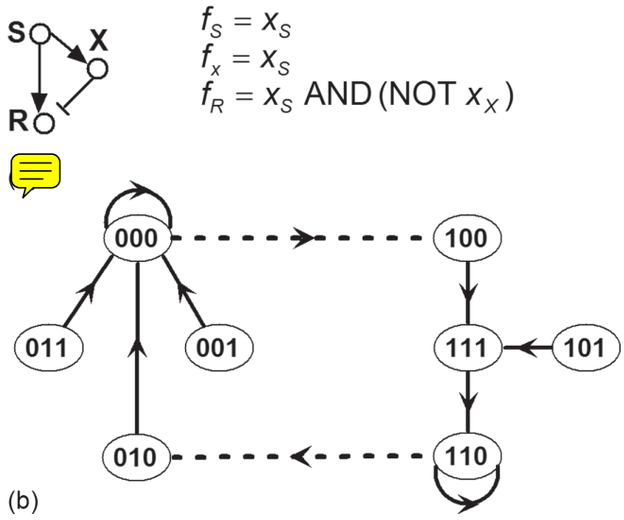

FIGURE 4.10 The Boolean correspondent of the model in the top row of Figure 4.9. (a) The network and Boolean transition functions. (b) The state transition graph of the model. The node states are shown in the order S, X, R. The transitions shown as dashed lines correspond to an externally set change in the value of the signal variable x_S. The path that starts from the 000 steady state and ends in the 110 steady state qualitatively reproduces the excitation-adaptation behavior of x_R seen in the continuous model.

Exercise 4.16. Consider the network in Figure 4.5a.

1. Does it have a commonality with the network in Figure 4.10? Explain.
2. Consider the transition function $f_C = x_B$ OR (NOT x_A) for node C. Using the synchronous state transition graphs calculated in Exercise 4.6, determine the trajectory of the system after x_A undergoes a step increase from state 001. Is there an excitation-adaptation behavior in x_C? □

Multistability is a phenomenon that arises often in physics, biology, and chemistry. Simply put, multistability is the ability of a system to achieve multiple steady states under the same external conditions. When there are two such states, we talk about *bistability*. In biology, bistability plays a key role in many fundamental processes such as cell division, differentiation, gene expression, cancer onset, and apoptosis. The example in [25] reproduced in the bottom row of Figure 4.9 illustrates that a signal-driven positive feedback loop can lead to bistability. The equation for the concentration of protein R in this model is

$$\frac{\mathrm{d}R}{\mathrm{d}t} = k_0 EP\,(R) + k_1 S - k_2 P$$

where $EP(R)$ is a sigmoidal function shown as the lowest dashed curve in the bottom middle of Figure 4.9, and k_i, $i = 0 \ldots 2$ are rate constants. This model leads to an irreversible switch from the low-value steady state of R to the high-value one at a critical value of the signal (see bottom right panel of Figure 4.9). This leads to a history-dependent behavior (*hysteresis*): when the signal is gradually increased from zero, the steady state concentration of R increases on the curve corresponding to the lower-value stable steady state, then switches to the higher curve when S goes beyond the critical value. If we now gradually decrease the signal back to zero, the steady state value of R stays on the upper curve.

Example 4.5. Let's see if a Boolean model can recapitulate this memory-dependent behavior. First, note that the sigmoidal nature of EP as a function of R lends itself easily to a Boolean approximation. We could pick a value R_T (T for threshold) such that the value of the function $EP(R_T)$ is about one-half the size of the "jump." We'll consider $EP(R)$ to be 0 for values of $R < R_T$ and 1 for values of $R \geq R_T$. With this, we can think of EP as a node in the Boolean network that is influenced by R and, in turn, it influences R. Because the function $EP(R)$ is increasing, the node R influences the node EP in a positive way. In Figure 4.7 and associated Exercise 4.10 we have seen a Boolean model of a network having an input node (signal) and a positive feedback loop. Inspecting

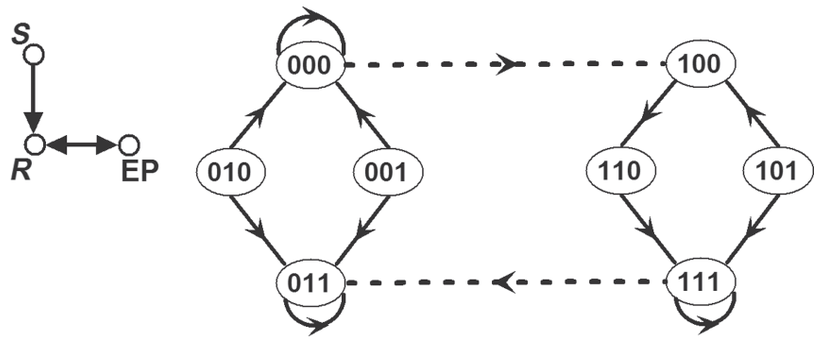

FIGURE 4.11 The Boolean correspondent of the model in the bottom row of Figure 4.9. The transitions shown as dashed lines correspond to an externally set change in the value of the signal variable x_S. For simplicity the loops that correspond to single node updates that do not change the node's state are not shown in the state transition graph, only loops that correspond to fixed points. The trajectory from the state 000 through 100, 110, 111, and 011 qualitatively reproduces the hysteresis of the continuous model. The order of variables is S, R, EP.

the regulation of R in the continuous model, we see that its synthesis is catalyzed independently by S and EP, thus an OR rule is the appropriate choice. Thus the model in Figure 4.7, with a suitable renaming of the node names, corresponds to the continuous model (see Figure 4.11). The transition functions are

$$f_S = x_S$$
$$f_R = x_S \quad \text{OR} \quad x_{EP}$$
$$f_{EP} = x_R$$

Let's consider general asynchronous update (the reader is encouraged to show as an exercise that using synchronous update gives similar results). As we have seen in Exercise 4.10, this model yields two fixed-point attractors when the signal is OFF, and a single attractor when the signal is ON. The continuous model also had two stable steady states for low values of the signal, and only one (the higher-value one) for signal levels above the critical level. Now consider that the system is initially in the steady state 000, and let's increase x_S to 1. The state 100 is not a steady state, and the system converges into the steady state 111. Thus the steady state value of R has switched from 0 to 1. Now let's decrease x_S to 0. The state 011 is a steady state, and the system remains there. Thus the steady state value of R did not go back to 0. We can conclude that the Boolean model qualitatively reproduces the hysteresis of the continuous model. □

Exercise 4.17. Consider the network in Figure 4.5b.

1. Does this have a commonality with the network in Figure 4.11? Explain.
2. Consider the transition function $f_B = x_A$ OR (NOT x_C) for node B. Using the general asynchronous state transition graphs calculated in Exercise 4.8, determine the trajectory of the system when x_A is switched to 1 from steady state 001, then switched to 0 in steady state 110. Does this system exhibit hysteresis? □

4.8 HOW TO DEAL WITH INCOMPLETE INFORMATION WHEN CONSTRUCTING THE MODEL

In the discussion so far, we have assumed that the entire signaling network is well known and well understood, with no knowledge gaps regarding its structure and mechanisms of interaction. In reality, however, there is often a need to deal with limited or incomplete experimental information while building a model. The following situations are typical: (1) not all links in network topology may be known with certainty; (2) when it is known that two or more nodes influence a node, the exact nature of the influence may still be unclear; (3) the initial condition for the system, which, ideally, should correspond to a relevant biological state, may not be known

a priori; and (4) when detailed information on the biomolecular kinetics of the signal transduction is lacking, deciding on an update algorithm may be challenging. We now briefly address each of these cases and outline some possible remedies.

4.8.1 Dealing with Gaps in Network Construction

There are two major types of causal experimental evidence from which information on edges of a network can be extracted: physical or biochemical evidence indicating direct interaction between two components, and evidence of the effect of the genetic mutation or pharmacological inhibition of a particular component on another component. The latter evidence indicates a causal relationship between the two components, which may be due to a direct interaction or to a relationship mediated by other components. The integration of the indirect causal evidence is often challenging, as each such apparent pair-wise relationship may in fact reflect a set of adjacent edges (a path) in the network, and it may involve other known or unknown nodes. In some cases, evidence from multiple experiments yields composite causal relationships, which then need to be broken down to component-to-component relationships, depending on the concrete situation.

Example 4.6. Consider a hypothetical signal transduction network with input node I and output node O which includes two known mediators, $M1$ and $M2$, and an unknown number of so-far unidentified mediators. Assume that experimental inhibition of mediator $M1$ has led to the conclusion that $M1$ is a positive regulator of the signal transduction process. This can be schematically represented by drawing an edge between I and O (to stand for the whole signal transduction process) and also drawing an edge that starts from $M1$ and points to the edge between I and O. A similar experiment has led to the conclusion that $M2$ is also a positive regulator of the signal transduction process, thus we can draw an edge that starts from $M2$ and points to the edge between I and O. A third experiment has indicated that the up-regulation of I leads to the up-regulation of $M1$. This is represented by a positive edge directed from I to $M1$. A fourth experiment has led to an edge starting from $M1$ and ending in $M2$. Figure 4.12a shows the resulting network, which is not yet in the form of a graph, as there are edges pointing to edges.

Our goal now is to find the most parsimonious network graph consistent with the combined experimental evidence reflected in Figure 4.12a. To transform this into a graph, let us interpret the edge that starts at $M1$ and points to the $I \rightarrow O$ edge as $M1$ activating an unknown node situated between I and O. Let's represent this unknown node with a black dot. Similarly, we can interpret the edge that starts at $M2$ and points to the $I \rightarrow O$ edge as $M2$ activating a second unknown node situated between I and O (Figure 4.12b).

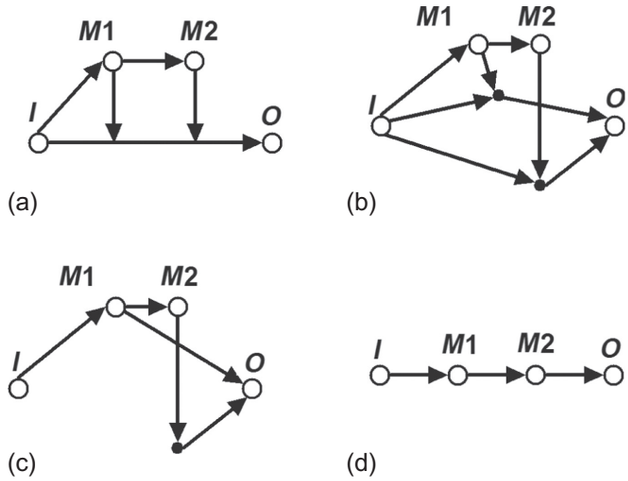

FIGURE 4.12 Illustration of the interpretation and simplification of causal information in order to find the most parsimonious signal transduction network. The interpretation involves the addition of unknown mediators, here shown as black dots. The simplification steps include transitive reduction and the collapsing of unknown mediator nodes.

Because the edges of this network correspond to causal effects but not to actual interactions, some of them are redundant as longer paths also express the same causal effect. For example, the edge between I and the upper unknown node (upper black dot) is explained by the two-edge path mediated by $M1$. Similarly, the edge between I and the lower black dot is explained by the three-edge path mediated by $M1$ and $M2$. These redundant edges can be deleted; this process is called *transitive reduction*. The upper black dot now has one incoming edge (from $M1$) and one outgoing edge (to O), thus it does not add any new causal information. For this reason, it can be eliminated by merging it with $M1$ (or alternatively with O), leading to a direct edge between $M1$ and O. Figure 4.12c shows the current incarnation of the network. Notice that we can now do another step of transitive reduction by deleting the edge between $M1$ and O, because it is explained by the $M1 \to M2 \to \bullet \to O$ path. Finally, we can compress the black dot between $M2$ and O, yielding the linear network shown in Figure 4.12c. In summary, the most parsimonious explanation of the four causal relationships is a linear path between I and O in which $M1$ is first and $M2$ is second. □

The approach illustrated in Example 4.6 was used to construct a model of drought-induced signal transduction in plants [15], specifically, the closure of stomata (microscopic pores on the leaves) in response to the drought hormone abscisic acid. We have seen a part of this network in Figure 4.3. Li et al. [15] collected more than 140 causal relationships derived from more than 50 literature citations on the regulation of stomatal closure by abscisic acid. The majority of these relationships were of type "C promotes the process (A promotes B)." The network resulting from the synthesis, interpretation, and simplification of these relationships had 54 nodes and 92 edges. The method was later formalized by DasGupta et al. [32, 33] and implemented in the software NET-SYNTHESIS. The software and its documentation can be downloaded from http://www2.cs.uic.edu/~dasgupta/network.synthesis/. The best use of this software is in iteration with additional literature search until the most appropriate network representation of the available experimental observations is found. This software can be used to construct the most parsimonious network consistent with a set of causal experimental evidence or to simplify an existing directed network in such a way that the causal relationships are preserved.

Exercise 4.18. Consider the list of causal evidence shown in Table 4.2. Use the software NET-SYNTHESIS to construct the most parsimonious signal transduction network. □

Exercise 4.19. Consider the network specified by the list of edges in Table 4.3. Use the software NET-SYNTHESIS to simplify this network by designating the nodes TCR, PDGFR, NFKB, and Caspase as pseudo-nodes and merging pseudo-nodes with regular nodes. An easy way to designate a node as pseudo-node in NET-SYNTHESIS is to precede its name by * (e.g., *TCR), either in the input file or by right-clicking on the node name in the displayed network. □

4.8.2 Dealing with Gaps in Transition Functions

We have already noted several times that for nodes with multiple regulators the knowledge of the incoming edges (positive and negative regulators) does not uniquely determine the dependency relationships among the node states. Thus, even complete knowledge of the networks does not by itself contain enough information to determine the transition rules for the nodes with multiple incoming edges. Assume, as an example, we know that two nodes A and B regulate a third node C. This could be an AND regulation (that is, both A and B would need to be ON to turn C ON) or it could be an OR relationship (when it would be enough for either one of A and B to be ON to turn C ON). Thus, additional information is needed.

One way to deal with this problem is to *knock out* one of the regulators, A or B, then examine the effect on C. In genetics, "gene knockout" is a technique used to make a gene inoperative. The term knockout here is used in the same sense: knockout of node A means setting and maintaining $x_A = 0$. If C remains permanently OFF after knockout of A, that would mean that both A and B are needed to turn C ON (thus, the transition function of C would be $f_C = x_A$ AND x_C); if not, we could conclude that the activation of C requires A or B, corresponding to the transition function $f_C = x_A$ OR x_C. In case of more than two regulators, the process also begins with examining the effect of knockout of one of the regulators on the state of the target node. If the information is still insufficient, several versions should be constructed and compared.

TABLE 4.2 A List of Causal Evidence Representative of What Could Be Synthesized from the Experimental Literature

Source Node	Causal Effect	Target Node or Edge	Direct Interaction?
ABA	Activates	InsPK	No
ABA	Activates	NO	No
ABA	Activates	PLD	No
NO	Activates	CIS	No
PA	Activates	ROS	No
ROS	Activates	CaIM	No
Ca^{2+}_{C}	Activates	AnionEM	No
Ca^{2+}_{C}	Activates	NO	No
PLD	Activates	PA	Yes
CIS	Activates	$Ca^{2+}{}_{c}$	Yes
CaIM	Activates	$Ca^{2+}{}_{c}$	Yes
AnionEM	Activates	Closure	Yes
KOUT	Activates	Closure	Yes
InsPK	Activates	ABA → CIS	No
Ca^{2+}_{ca}	Activates	NO → AnionEM	No
CaIM	Activates	ABA → KOUT	No

This list is derived from the work in [16] and has the same node names, but it is much simpler than the original.

Example 4.7. Consider the network in Figure 4.8, and assume the Boolean transition functions are unknown. Because A is the signal to the network, its transition function is clear: $f_A = x_A$. Inspecting the figure, the transition function for node B depends on x_A and on (NOT x_C), while the transition function of node C depends on (NOT x_A) and on (NOT x_B). Thus the possibilities are:

$$f_B = x_A \text{ OR } (\text{NOT } x_C)$$
$$f'_B = x_A \text{ AND } (\text{NOT } x_C)\,;$$
$$f_C = (\text{NOT } x_A) \text{ OR } (\text{NOT } x_B)$$
$$f'_C = (\text{NOT } x_A) \text{ AND } (\text{NOT } x_B)$$

Let's now imagine that we search the literature and find evidence for a steady state in which $x_A = 1$, $x_B = 1$ and $x_C = 0$. Does this information eliminate any of the candidate transfer functions?

Plugging in those state values we obtain $f_B = 1$, $f'_B = 1$, $f_C = 0$, $f'_C = 0$. Both transfer function variants give the same result, which is in agreement with the node's steady states; thus, this information did not help limit the possibilities.

Imagine now that after more search we find that in the case when $x_A = 1$ and simultaneously node B is knocked out (i.e., x_B is set to 0), the steady state of C is 0. In this case $f_C = 1$ and $f'_C = 0$, and only the latter is consistent with the observed $x_C = 0$. Thus, we should conclude that the transfer function of node C is $f'_C =$ (NOT x_A) AND (NOT x_B).

Finally, let's assume that a third observation indicates that in the case when $x_A = 0$ and simultaneously node C is knocked out, the steady state of B is 1. In this case $f_B = 1$ and $f'_B = 0$, of which only the former is consistent with the observed steady state. Thus, we conclude that the transfer function of node B is $f_B = x_A$ OR (NOT x_C). □

TABLE 4.3 The List of Edges in a Two-Signal, One Output Signal Transduction Network Used in Exercise 4.19

Source Node	Causal Effect	Target Node
Stimuli	Activates	TCR
TCR	Activates	RAS
PDGF	Activates	PDGFR
S1P	Activates	PDGFR
PDGFR	Activates	S1P
RAS	Activates	FAS
S1P	Inhibits	FAS
FAS	Inhibits	S1P
FAS	Inhibits	NFKB
FAS	Activates	Caspase
NFKB	Inhibits	Caspase
Caspase	Activates	Apoptosis

This network is derived from the T cell apoptosis signaling network displayed in Figure 4.2, and has the same node names, but it is much simpler than the original.

The approach illustrated in Example 4.8 was used in [34] to construct and refine Boolean models from an initial signal transduction model by calibrating the model against measurements of protein abundance or activity. The initial signal transduction model was first simplified to collapse nodes for which no measurements were available, in a way similar to the collapsing of pseudo-nodes we saw in Example 4.6 and Exercise 4.19. Then an ensemble of models was generated from every possible transition function consistent with the network. Finally, those models were evaluated by comparing their steady states with the experimental observations and the most consistent and also most parsimonious model was selected. Application of this method to the signal transduction network that mediates early signaling downstream of seven cytokine and growth factor receptors in human liver cells, using measurements of sixteen proteins in this network, led to significant refinement (mostly edge deletion, but also a few additions) of an original database-derived network. The final network and Boolean model was validated by follow-up experimental measurements. This method is instantiated in the freely available software package CellNOpt [35].

Exercise 4.20. Consider the networks in Figure 4.5.

1. Assume that for the network in Figure 4.5a, an experimental observation is consistent with the steady state $x_A = x_B = x_C = 1$. What transition function does this imply for node C?
2. Assume that for the network in Figure 4.5b, an experimental observation is consistent with the steady state $x_A = x_C = 1$ and $x_B = 0$. What transition function does this imply for node B? □

4.8.3 Dealing with Gaps in Initial Condition

Ideally, the model's starting state should be the biologically relevant resting or pre-stimulus state, if it is known *a priori*. If the available information is insufficient, one can enumerate or, if that is difficult, sample a large number of initial conditions wherein certain nodes are in a known state while the state of others can vary. A large number

of replicate simulations should be done, and the results need to be summarized over these replicate simulations. For example, one calculates the fraction of realizations of a certain attractor. We can think of these replicate simulations as a population of cells which differ in their pre-stimulus states, and the fraction of realizations of an attractor can be interpreted as the probability that the system attains the corresponding cellular phenotype. This approach was used, e.g., by Li et al. in [15] in the context of constructing a model of abscisic acid-induced closure. Because many of the nodes of this network are also involved in the response to other signals, for example, light and atmospheric CO_2, it is difficult to estimate their state prior to receiving the abscisic acid signal. Thus, the initial state of 38 intermediary (non-source, non-sink) nodes was randomly chosen, and 10,000 replicate simulations were performed. The authors studied the fraction of simulations in which the output node Closure was ON, and found that it reached 1 after eight updates of all the nodes (using random order asynchronous update). Thus, they were able to conclude that the initial state of the intermediary nodes did not affect the steady state of the node Closure.

Exercise 4.21. What initial conditions should be considered for Example 4.1 if we are interested in the system's response to a sustained signal? □

Exercise 4.22. How many initial states should be considered for an N-node network if we have no information on the actual initial state? □

4.8.4 Dealing with Gaps in Timing Information

We can choose an updating scheme that is most realistic for the biological system of interest, or compare different schemes with the same system. In cases where there is no information to guide the choice of update scheme, updating one node at a time (general asynchronous update) is the most parsimonious choice because its results are also representative of the random order update [16]. The biological system of interest for the study [16] was the signal transduction network by which plants respond to the drought hormone abscisic acid, first modeled in [15]. The study found that the state of the majority of the nodes, including Closure, stabilized regardless of the updating scheme used. A subset of nodes regulated by cytosolic Ca^{2+} had one or two different behaviors: fluctuations that eventually decayed, leading to a steady state in which cytosolic Ca^{2+} was OFF, and sustained oscillations if and only if strict relationships among the timing of Ca^{2+} production and decay were satisfied. There is evidence in the experimental literature for abscisic acid induced oscillations in Ca^{2+}, but not enough to establish whether they are sustained or not. Likewise, the timing or kinetics of Ca^{2+} production and decay are not known. However, the model suggested that a transient increase in Ca^{2+} was sufficient for a successful closure response and sustained oscillations were not necessary. We encourage the reader to examine this article as a way to develop further understanding of the effect that different updating schemes may have on the long-term behavior of Boolean models.

4.9 GENERATE NOVEL PREDICTIONS WITH THE MODEL

A Boolean model can be used to analyze the changes in the system's attractor repertoire in the case of system perturbations. Knockout of a component can be simulated by fixing the corresponding node in the OFF state; constitutive expression can be simulated by fixing the node's state as ON. Transient perturbations can also be studied by implementing temporary (reversible) changes to the node's states. The model can predict the changes in the attractors of the system and their basins of attraction and identify the perturbations that lead to dramatic changes. This way, perturbation analysis can identify key components that are essential to phenotype traits [5, 6, 15]. If the studied phenotype corresponds to a disease, the identified essential components are candidate targets for therapeutic interventions.

Exercise 4.23. Consider Example 4.1 (shown in Figure 4.4) when node B is knocked out (i.e., x_B is set to 0). What is the relevant state space now? Construct the state transition graph corresponding to synchronous update and general asynchronous update. Compare with Figure 4.13. □

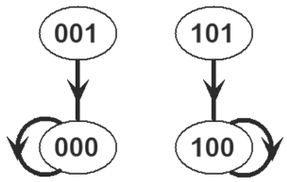

FIGURE 4.13 State transition graph for Example 4.1 (the network in Figure 4.4) when node B is knocked out. Because nodes A and B are not updated (A is a sustained signal and B is knocked out), synchronous and general asynchronous update are equivalent in this case.

Exercise 4.24. Compare Figure 4.6 with Figure 4.13. How did the steady states of the system change due to the knockout of node B? □

Exercise 4.25. Determine the state transition graph of the network in Example 4.1 (Figure 4.4) for synchronous update when node C is knocked out. □

Exercise 4.26. Determine the state transition graph of the network in Figure 4.4 for general asynchronous update when node C is knocked out. □

Exercise 4.27. Compare Figure 4.6 to your results in Exercises 4.25 and 4.26. How did the steady states of the system change due to the knockout of node C? □

An example of perturbation analysis is the study of nodes whose knockout can impair abscisic acid-induced closure [15, 16]. Using stochastic asynchronous update, the unperturbed system had a single fixed-point attractor which included the ON state of the node Closure. Systematic study of each intermediary node's knockout led to the identification of three perturbation categories: knockouts that led to the normal steady state (which represented 75% of all knockouts), knockouts leading to a steady state in which Closure was OFF (22.5% of all knockouts), and a single knockout, which represents the clamping of the cytosolic pH level, leading to a complex attractor in which Closure fluctuated between ON and OFF. Thus, one could conclude that the signal transduction process was robust to the large majority of perturbations, but also sensitive to the impairment of a few key nodes. These key nodes should be studied further to establish the ways in which their perturbations could be prevented. □

Exercise 4.28. Consider the model of Example 4.3. As shown in Figure 4.7b, the system has two steady states, 000 and 011, when the signal is OFF ($x_I = 0$). Under general asynchronous update, both steady states are reachable from the initial conditions 001 and 010. Let's assume that steady state 000 is undesirable. Can you find a state manipulation (fixing the state of a node) such that state 000 becomes unreachable? □

As we have seen in Exercise 4.17, mutual inhibition among two nodes can lead to the same behavior as mutual activation between the same nodes. Mutual inhibition between two groups of nodes is in fact a key feature of the T-cell apoptosis signaling network (see Exercise 4.19). Indeed, as in Example 4.3, the Boolean model of this network has two steady states, one corresponding to apoptosis and one corresponding to an abnormal cell fate seen in T-LGL leukemia [5, 6]. A state manipulation that could potentially eliminate this latter steady state is to fix a node's state in the opposite state that it stabilizes in the T-LGL steady state. Considering this manipulation for each intermediary node in the network led to the identification of nineteen potential therapeutic targets for the disease [6]. More than half of these manipulations were supported by available experimental data or by follow-up experiments, and the rest can guide future experiments.

4.10 BOOLEAN RULE-BASED STRUCTURAL ANALYSIS OF CELLULAR NETWORKS

Analysis of all relevant dynamic trajectories of a system that is bigger than the simple examples we have considered here is complex and time consuming. The good news is that sometimes important conclusions can be drawn without dynamic simulations, based on graph-theoretic analysis alone. A first step in this direction is to resolve the ambiguity pertaining to nodes with multiple regulators by incorporating the Boolean rules into the

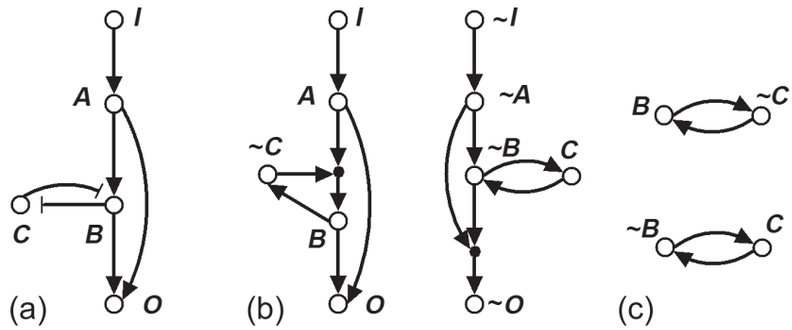

FIGURE 4.14 Illustration of methods that integrate the structure and logic of regulatory interactions. (a) A hypothetical signal transduction network. (b) The expanded representation of the network which integrates the Boolean transition functions $f_B = x_A$ AND NOT x_C, and $f_O = x_A$ OR x_B. The expanded network includes five complementary nodes, indicated by preceding the node name by $\sim$, and two composite nodes, indicated by small black filled circles. (c) The stable motifs of the expanded network in the case of a sustained input signal ($x_I = 1$). The first stable motif corresponds to the state 11101 (in the order I, A, B, C, O), while the second stable motif corresponds to the state 11011.

network topology [36]. Specifically, one introduces a *complementary node* for each node of the network, and a *composite node* for each set of interactions with conditional dependency. We illustrate the process with the next example.

Example 4.8. Consider a signal transduction network composed of the input node I, intermediary nodes A, B, and C and the output node O (Figure 4.14a). This network shares some features of the core T-LGL network derived in [6]. The network does not completely specify the transition functions of nodes B and O. Let's specify the transition functions as $f_B = x_A$ AND (NOT x_C), and $f_O = x_A$ OR x_B. The complete set of transition functions now is:

$$\begin{aligned} f_A &= x_I \\ f_B &= x_A \text{ AND } (\text{NOT } x_C) \\ f_C &= \text{NOT } x_B \\ f_O &= x_A \text{ OR } x_B \end{aligned}$$

Let's now construct the expanded network that integrates the transition functions. The expanded network features the addition of a complementary (negated) node for each real (original) node in the system, denoted by preceding the real node's name with $\sim$. The state of this node is the negation of the state of the corresponding real node, and the transition function of the negated node is the logic negation of the transition function of the original node. For example, the transition function of the complementary node $\sim A$ is $f_{\sim A} = \text{NOT } x_I = x_{\sim I}$, indicating that $\sim A$ is positively regulated by the complementary node $\sim I$. The transition function of the complementary node $\sim B$ is $f_{\sim B} = \text{NOT } (x_A \text{ AND } (\text{NOT } x_C)) = (\text{NOT } x_A) \text{ OR } x_C$. This means that $\sim B$ is positively regulated by $\sim A$ and C.[1] The transition function for $\sim O$ is $f_{\sim O} = \text{NOT } (x_A \text{ OR } x_B) = (\text{NOT } x_A) \text{ AND } (\text{NOT } x_B)$. The complete set of transition functions for complementary nodes is now

$$\begin{aligned} f_{\sim A} &= \text{NOT } x_I \\ f_{\sim B} &= (\text{NOT } x_A) \text{ OR } x_C \\ f_{\sim C} &= x_B \\ f_{\sim O} &= (\text{NOT } x_A) \text{ AND } (\text{NOT } x_B) \end{aligned}$$

Next, we introduce composite nodes. For this the transition functions need to be specified in a *disjunctive normal format*, meaning that AND clauses are grouped and separated by OR's. For example, the expression (A AND B) OR (C AND D) is in a disjunctive normal form, while the expression A AND (B OR C) is not.

1. Recall the De Morgan laws for Boolean expressions: (1) NOT (A AND B) = (NOT A) OR (NOT B) and (2) NOT (A OR B) = (NOT A) AND (NOT B).

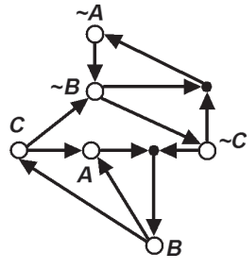

FIGURE 4.15 An expanded network for Exercise 4.29.

Inspecting the transition functions, we can verify that they are in the disjunctive normal format. We now add a composite node for each AND clause in the transition functions. Specifically, there will be a composite node for the expression x_A AND (NOT x_C), which regulates B, and another one for (NOT x_A) AND (NOT x_B), which regulates the complementary node $\sim O$.

The expanded network is shown on Figure 4.14b. Note that the update rules for all nodes with multiple inputs are now uniquely determined from the topology of the expanded network. All multiple inputs for a composite node are of type AND, while for the rest of the nodes multiple dependencies are of type OR. □

Exercise 4.29. Consider the expanded network in Figure 4.15. Construct the transition functions of the nodes A, B, C. Construct the transition functions of the complementary nodes $\sim A$, $\sim B$, $\sim C$. Verify that the transition function of each complementary node is the logic negation of the transition function of the respective original node. □

In Example 4.8 and Figure 4.14b, the expanded network is composed of two components that are disconnected from each other. The first component starts with the input node I, ends in the output node O, and contains A, B, $\sim C$ and a composite node shown as a black dot. The second component is made up by four complementary nodes, node C, and a second composite node. We can interpret the two subgraphs as the information transmission networks corresponding to the presence of the signal (left) and to the absence of the signal (right). The fact that the left subgraph contains both the input and output node indicates that there is at least one path that connects the input and output of the system (the signal and the system's response). The shortest such path is I, A, O. The next shortest is I, A, composite node, B, O. Is this path a context-independent conduit of information, or does its success depend on other nodes?

Because it involves a composite node, which stands for an AND clause embodying conditionality, this path is, in fact, dependent on the complementary node ∼C. Only a subgraph that contains all regulators of a composite node can serve as an independent information propagation conduit. This concept was termed an *elementary signaling mode* in [36] and is defined as the minimal set of components able to perform signal transduction independently. Thus, the network in Figure 4.14 contains two elementary signaling modes between input node I and output node O: the path I, A, O and the subgraph that contains I, A, the composite node, $\sim C$, B and O. Both of these are minimal because taking a node or edge away would obstruct the propagation of the signal.

The elementary signaling modes can be used to quantify the importance of nodes in mediating the signal. For example, in Figure 4.14b the loss of node A eliminates both elementary signaling modes between I and O, but the loss of node B leaves one of them intact. Application to several signaling networks, including those for abscisic acid-induced closure [15] and for T-cell apoptosis signaling [5], showed that nodes whose loss disrupts all elementary signaling modes were also essential to the model's dynamic attractor(s), in the sense that their knockout made this attractor unreachable [6, 36]. These results indicate that elementary signaling mode analysis, a method that involves Boolean logic and graph theory but no dynamic simulations, can be effectively used as a preliminary to or even as a substitute for dynamic perturbation analysis.

Exercise 4.30. Consider the model of Figure 4.14a in the case of a sustained signal ($x_I = 1$).

1. Determine the attractors of the system under general asynchronous update. Which of these attractors corresponds to a response to the signal?
2. Set node A to OFF. Determine the attractors of the system. Did at least one attractor remain that corresponds to a response to the signal? What is your conclusion? Is node A essential for the signal transduction process?
3. Set node B to OFF. Did at least one attractor remain that corresponds to a response to the signal? What is your conclusion? Is node B essential to the signal transduction process?
4. Let's assume that the ON state of the output node ($x_O = 1$) is undesirable. What node interventions would make this outcome impossible? □

The expanded network can also be used as a basis for network simplification. As shown in [37], a topological criterion can be used to identify network motifs (subgraphs) that stabilize in a fixed state regardless of the rest of the network. A *stable motif* in the expanded network is defined as the smallest strongly connected component (SCC) with the following properties: (1) the SCC cannot contain both a node and its complementary node and (2) if the SCC contains a composite node, it also contains all of its input nodes. For example, in Figure 4.14b the nodes $\sim B$ and C form a stable motif. The fixed state of the nodes in the stable motif can be directly read out from the expanded network: if the stable motif contains the node, the node stabilizes in the ON state, and if the stable motif contains a complementary node, the corresponding node stabilizes in the OFF state. These fixed states can be plugged into the transition functions of other nodes, leading to simpler functions, and consequently to a simpler expanded network. Iterative searching for stable motifs and network simplification leads to one of two possible outcomes: either there are no nodes with unknown states, in which case a fixed point of the system is identified, or no new stable motifs are found, in which case the remaining nodes are expected to oscillate. Thus stable motif analysis serves as a preliminary to or as a substitute for attractor analysis. For example, the T-cell apoptosis signaling network has four stable motifs. Iterative simplification of these stable motifs leads to the system's steady states, the same as those found from dynamic simulations. Interestingly, the stable motif formed by three nodes close to the PDGF signal can solely determine the steady state, regardless of the other motifs or the trajectory of the system leading up to the stabilization of the motif [37].

Overall, these integrated structural and logic methods are fruitful as exploratory analysis of large signaling networks where dynamic modeling is computationally impractical, or as a first step that guides follow-up targeted computational or experimental studies.

Exercise 4.31. Consider the sustained presence of the input signal ($x_I = 1$) in Figure 4.14a. Simplify the transition functions of the nodes and construct the expanded network corresponding to this case. Compare with Figure 4.14c. □

Exercise 4.32. Determine the stable motifs of the expanded network in Figure 4.14c. Compare with the steady states found in Exercise 4.30. □

Exercise 4.33. Consider again the network in Figure 4.14a, but this time consider the following update rules for nodes B and C: $f_B = x_A$ OR (NOT x_C), and $f_O = x_A$ AND x_B.

1. Construct the expanded network.
2. Determine the elementary signaling modes between input node I and output node O in the expanded network.
3. Determine the essential signal-mediating nodes based on the elementary signaling modes.
4. Consider the sustained absence of the signal $x_I = 0$. Determine the expanded network, its stable motifs, and the corresponding steady states. □

Exercise 4.34. Consider the network in Figure 4.16. Construct two sets of transition functions that are consistent with this network. For each set,

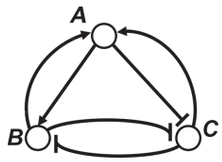

FIGURE 4.16 A simple three-node network for Exercise 4.34.

1. Construct the expanded network.
2. Determine the stable motifs in the expanded network and the corresponding steady states.
3. Verify your results for part 2 by determining the model's steady states analytically. □

4.11 CONCLUSIONS

Although Boolean network models have a limited capacity to describe the quantitative characteristics of dynamic systems, they do exhibit considerable dynamic richness and have proven effective in describing the qualitative behaviors of signal transduction networks, in predicting key components, and in proposing effective intervention strategies. The fact that Boolean models do not require the knowledge of kinetic parameters makes them a preferred choice for systems where these parameters have not been measured. Thus Boolean models pass the two key tests: they are useful, and they help us to better understand the systems for which they are formulated. The success of Boolean models illustrates that in at least a subset of biological systems the organization of network structure plays a more important role than the kinetic details of the individual interactions. Thus, Boolean networks can serve as a foundation for the modeling of signaling networks, upon which more detailed continuous models can be built as kinetic information and quantitative experimental data become available. The simpler Boolean model can be used for efficient exploratory analysis to fix the model's structure and to help develop a refined continuous model, which in turn can be further compared with quantitative biological observations.

REFERENCES

[1] Gomperts BD, Tatham PER, Kramer IM. Signal transduction. San Diego, CA: Academic Press; 2002.
[2] Gordon KJ, Blobe GC. Role of transforming growth factor-beta superfamily signaling pathways in human disease. Biochim Biophys Acta 2008;1782:197–228.
[3] Ikushima H, Miyazono K. TGFbeta signalling: a complex web in cancer progression. Nat Rev Cancer 2010;10:415–24.
[4] Albert R, Wang RS. Discrete dynamic modeling of cellular signaling networks. Methods Enzymol 2009;467:281–306.
[5] Zhang R, Shah MV, Yang J, Nyland SB, Liu X, Yun JK, et al. Network model of survival signaling in large granular lymphocyte leukemia. Proc Natl Acad Sci U S A 2008;105:16308–13.
[6] Saadatpour A, Wang RS, Liao A, Liu X, Loughran TP, Albert I, et al. Dynamical and structural analysis of a T cell survival network identifies novel candidate therapeutic targets for large granular lymphocyte leukemia. PLoS Comput Biol 2011;7:e1002267.
[7] Albert R, Barabasi AL. Statistical mechanics of complex networks. Rev Mod Phys 2002;74:47–97.
[8] Gilmore TD. Introduction to NF-kappaB: players, pathways, perspectives. Oncogene 2006;25:6680–4.
[9] Alon U. An introduction to systems biology: design principles of biological circuits. Boca Raton, FL: Chapman and Hall/CRC; 2006.
[10] Thomas R, d'Ari R. Biological feedback. Boca Raton, FL: CRC Press; 1990.
[11] Ma'ayan A, Jenkins SL, Neves S, Hasseldine A, Grace E, et al. Formation of regulatory patterns during signal propagation in a mammalian cellular network. Science 2005;309:1078–83.
[12] Smoot ME, Ono K, Ruscheinski J, Wang PL, Ideker T. Cytoscape 2.8: new features for data integration and network visualization. Bioinformatics 2011;27:431–2.
[13] Hagberg A, Schult D, Swart P. Exploring network structure, dynamics, and function using NetworkX. In: Varoquaux G, Vaught T, Millman J, editors. SciPy Conference, Pasadena, CA; 2008. p. 11–5.
[14] Batagelj V, Mrvar A. Pajek—analysis and visualization of large networks. In: Jünger M, Mutzel P, editors. Graph drawing software. Berlin: Springer; 2003. p. 77–103.

[15] Li S, Assmann SM, Albert R. Predicting essential components of signal transduction networks: a dynamic model of guard cell abscisic acid signaling. PLoS Biol 2006;4:e312.
[16] Saadatpour A, Albert I, Albert R. Attractor analysis of asynchronous Boolean models of signal transduction networks. J Theor Biol 2010;4:641–56.
[17] Kestler HA, Wawra C, Kracher B, Kuhl M. Network modeling of signal transduction: establishing the global view. Bioessays 2008;30:1110–25.
[18] Karlebach G, Shamir R. Modelling and analysis of gene regulatory networks. Nat Rev Mol Cell Biol 2008;9:770–80.
[19] Sobie EA, Lee YS, Jenkins SL, Iyengar R. Systems biology–biomedical modeling. Sci Signal 2011;4:tr2.
[20] Kauffman SA. The origins of order: self organization and selection in evolution. New York, NY: Oxford University Press; 1993.
[21] Aldridge BB, Saez-Rodriguez J, Muhlich JL, Sorger PK, Lauffenburger DA. Fuzzy logic analysis of kinase pathway crosstalk in TNF/EGF/insulin-induced signaling. PLoS Comput Biol 2009;5:e1000340.
[22] Chaouiya C. Petri net modelling of biological networks. Brief Bioinform 2007;8:210–9.
[23] Saadatpour A, Albert R. Boolean modeling of biological regulatory networks: a methodology tutorial. Methods 2013;62:3–12.
[24] Garg A, Di Cara A, Xenarios I, Mendoza L, De Micheli G. Synchronous versus asynchronous modeling of gene regulatory networks. Bioinformatics 2008;24:1917–25.
[25] Tyson JJ, Chen KC, Novak B. Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. Curr Opin Cell Biol 2003;15:221–31.
[26] Robeva RS, Hodge TL. Mathematical concepts and methods in modern biology: using modern discrete models. New York, NY: Academic Press; 2013.
[27] Albert I, Thakar J, Li S, Zhang R, Albert R. Boolean network simulations for life scientists. Source Code Biol Med 2008;3:16.
[28] Mussel C, Hopfensitz M, Kestler HA. BoolNet—an R package for generation, reconstruction and analysis of Boolean networks. Bioinformatics 2010;26:1378–80.
[29] Hinkelmann F, Brandon M, Guang B, McNeill R, Blekherman G, et al. ADAM: analysis of discrete models of biological systems using computer algebra. BMC Bioinform 2011;12:295.
[30] Yi TM, Huang Y, Simon MI, Doyle J. Robust perfect adaptation in bacterial chemotaxis through integral feedback control. Proc Natl Acad Sci U S A 2000;97:4649–53.
[31] Takeda K, Shao D, Adler M, Charest PG, Loomis WF, et al. Incoherent feedforward control governs adaptation of activated ras in a eukaryotic chemotaxis pathway. Sci Signal 2012;5:ra2.
[32] Dasgupta B, Albert R, Dondi R, Kachalo S, Sontag E, et al. A novel method for signal transduction network inference from indirect experimental evidence. J Comput Biol 2007;14:927–49.
[33] Kachalo S, Zhang R, Sontag E, Albert R, DasGupta B. NET-SYNTHESIS: a software for synthesis, inference and simplification of signal transduction networks. Bioinformatics 2008;24:293–5.
[34] Saez-Rodriguez J, Alexopoulos LG, Epperlein J, Samaga R, Lauffenburger DA, et al. Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction. Mol Syst Biol 2009;5:331.
[35] Terfve C, Cokelaer T, Henriques D, MacNamara A, Goncalves E, et al. CellNOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. BMC Syst Biol 2012;6:133.
[36] Wang RS, Albert R. Elementary signaling modes predict the essentiality of signal transduction network components. BMC Syst Biol 2011;5:44.
[37] Zanudo JG, Albert R. An effective network reduction approach to find the dynamical repertoire of discrete dynamic networks. Chaos 2013;23:025111.