Analyzing Pathways using SAT-based Approaches

SRI International, Menlo Park, CA 94025

Abstract. A commonly used language for representing knowledge about a biological process is that of *reaction networks*. How does one understand such generic networks and answer queries using them? In this paper, we present a novel approach based on translation of generic reaction networks to Boolean *weighted MaxSAT*. The Boolean weighted MaxSAT instance is generated by encoding the basic meaning of a reaction network by boolean clauses and assigning weights to these clauses based on the relative importance of the meaning they capture. The important feature of this translation is that it uses reactions, rather than the species, as the boolean variables. Existing weighted MaxSAT solvers are used to find maximum weight assignments for the generated clauses. This method of analyzing reaction networks is generic, flexible and scales to large models of reaction networks. We present a few case studies to validate our claims.

1 Introduction

A network of reactions is a convenient way to represent knowledge about a biological process. Each reaction converts some reactants into products in the presence of certain other molecules. A reaction can be modeled in various ways. If detailed information about its dynamics is available, then differential equations can be written that describe the dynamics of the reaction. Unfortunately, this information is not always available, especially in the case where the reaction is a biochemical reaction. Even when it is available, it is not available for each of the thousands of reactions that constitute a network.

In the absence of validated dynamical models, reaction networks can be studied as discrete models. Discrete models are obtained by abstracting time to a before-after relationship. In this context, a reaction can be seen as a discrete state transition, and a reaction network is then simply a discrete state transition system. For instance, a reaction can be thought of as a Petri net transition, and a reaction network as a Petri net.

A naive discrete model, wherein the state consists of the number of molecules of each species, has a horrendously huge state space. Any kind of analysis on such models is infeasible–both in theory and in practice. In theory, for instance, Petri net reachability is decidable, but with no known upper-bound. In practice, when analyzing systems containing just 100 total molecules of 4 different species, the state space size is 4^{100}.

To overcome the state space problem, several simpler models are considered. For instance, boolean models abstract species to being either *present* or *absent*.

Other qualitative abstractions, such as *absent*, *present in low quantities*, and *present in large quantities* are also possible.

In this paper, we present a new scalable approach for analyzing large reaction networks. There are three main features in our approach. First, it is based on qualitatively abstracting the reactions into two states–*on* and *off*. This is dual to the more conventional approach where the presence or absence of molecular species, and not *reactions*, is used to define the state of the system [4, 8, 9]. Second, it uses a boolean MaxSat as its backend engine. There is a generic translation from reaction networks to boolean MaxSat instances. Third, it is flexible. It performs several different types of analyses. Based on the type of analysis, different MaxSat instances are generated. Moreover, there is additional flexibility in that, apart from the boolean constraints generated for generic reaction networks, specific boolean constraints can be added for reaction networks encoding specific aspects, such as signaling pathways, or transcriptional regulation.

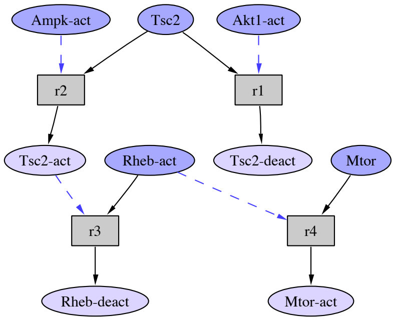

Fig. 1. A simple network with competing rules

Consider, for example, the very simple network[1] shown in Figure 1. This network consists of 4 reactions called ρ_1, ρ_2, ρ_3 and ρ_4. We will use this network as a running example in the paper.

$$\begin{array}{lrcl}
\rho_1: & \text{Tsc2} & \xrightarrow{\text{Akt-act}} & \text{Tsc2-deact} \\
\rho_2: & \text{Tsc2} & \xrightarrow{\text{Ampk-act}} & \text{Tsc2-act} \\
\rho_3: & \text{Rheb-act} & \xrightarrow{\text{Tsc2-act}} & \text{Rheb-deact} \\
\rho_4: & \text{Mtor} & \xrightarrow{\text{Rheb-act}} & \text{Mtor-act}
\end{array}$$

It is not immediately obvious how to understand even this simple network. Using the approach described in this paper, all possible "steady-state" behaviors of the above network can be computed. For this example, the tool computes two

[1] This is not necessarily biologically accurate.

possible behaviors. Either Akt-act is present, deactivating Tsc2, while Mtor gets activated by Rheb-act (Reactions 1 and 4 are “on”); or, Ampk-act is present, activating Tsc2, which in turn deactivates Rheb-act (Reactions 2 and 3 are “on”). The important point here is that the steady-state behavior is thought of a subset of reactions that can be consistently “on”, as opposed to the traditional viewpoint where steady-state refers to species reaching some equilibrium concentrations.

As mentioned earlier, our approach can perform different kinds of analyses. We can specify an initial dish consisting of some of the species and ask the tool to identify all possible behaviors constrained to the given initial dish. In the above example, if the initial dish only contains Tsc2, Ampk-act, Rheb-act and Mtor, then our tool identifies that the second and third reactions can be “on”, and that the other option, where reactions 2 and 4 are “on” is less likely. Similarly, target species can be specified, and the tool will generate paths (scenarios) that will produce the target species. Each such scenario will be assigned a weight indicating its relative likelihood.

1.1 Motivation

The definition of “steady-state” behavior we use in this paper is unusual. A steady-state is identified as a subset of reactions that can be consistently “on”. Traditionally, a steady-state is identified by the equilibrium concentrations of all species in the network. One of the strong motivations for this definition is that signaling pathways are best understood in this way. More than the individual species concentrations, it is the chain of reactions that capture how information flows from the cell membrance to effect downstream activities in a cell. This chain of reactions corresponds directly to the notion of a steady-state in our approach.

The different reactions in the steady-state chain of reactions will, in reality, be temporally separated. While certain phosphorylation activity may occur in a few minutes after a cell is hit by ligands, other downstream activities may occur much later. In our approach, we identify the whole chain as one possible steady-state behavior of the reaction network. The complete chain of reactions may never simultaneously be “on” in reality. However, they are still useful in understanding the function of a given complex reaction network.

The approach based on translation to MaxSat is motivated by the need for flexibility. Reaction networks have slightly different meaning in different contexts. Metabolic pathways, signaling pathways, and transcriptional regulation networks work on different notions of species and reactions. Our basic constraints attempt to capture the minimal common meaning that can be ascribed to any such network. The weights on the constraints give flexibility in making certain constraints harder than others in different contexts.

Finally, it should be mentioned that the technology for solving SAT and MaxSAT problems has made significant advances in recent years and problems with thousands of boolean variables and even more clauses are routinely solved in matter of seconds. We have used our tool on the HumanCyc database of metabolic pathways (containing over a thousand reactions) and we can asnwer queries in a few seconds.

2 Preliminaries

In this section, we formalize our terminology. A *species* is a generic name used to denote any entity, such as a molecule, ion, protein, enzyme, ligand, receptor, complex, or a postranscriptionally modified form of a protein. We do not differentiate between these different roles and just formally identify a species with a unique name. The set of all species will be denoted by S. A *reaction* consists of a set of *reactants*, a set of *modifiers*, and a set of *products*. Thus, a reaction ρ is a 3-tuple $\langle R, M, P\rangle$, where R, M, P are pairwise disjoint subsets of S. Given a reaction ρ, we denote its set of reactants, modifiers, and products by $R(\rho)$, $M(\rho)$, and $P(\rho)$ respectively.

A *network* $\mathcal{N}$ is a collection of reactions. A *network instance* is a network together with an optional set of *input* species, a set of *forbidden* species, and a set of *target* species.

A *pathway* is a special kind of network. Informally, a pathway contains a *related* set of reactions that can be consistently switched "on". The following sections will formally define the constraints we impose to identify pathways.

3 Biochemical Networks to Boolean SAT

In this section, we describe the procedure that generates a set of boolean constraints (clauses) from a network. Later in this section, we describe the additional constraints that are generated from a network instance. The idea behind the translation is that a solution (satisfying assignment) of the boolean constraints will represent a feasible behavior of the network (or network instance).

The key aspect of our boolean encoding is that we introduce a boolean variable for each reaction (and not for each species). The reason for this choice will be discussed later. If $\mathcal{N} = \{\rho_1, \rho_2, \ldots, \rho_n\}$ is a biochemical network, we generate boolean constraints over n boolean variables $b_1, \ldots, b_n$, where the i-th boolean variable b_i represents whether the i-th reaction ρ_i is "on" or "off". The boolean constraints are generated using a minimalistic approach: we encode only the very basic and least contentious interpretations of the reactions in $\mathcal{N}$ as boolean constraints.

3.1 Mutual Promotion

The first set of clauses encode the following rule:

> If a reaction is "on", then for each of its reactants (modifiers), there is atleast one reaction "on" that produces that reactant (modifier).

For a species $s \in S$, let *producers*(s) denote the set of reactions that have s in their set of products.

$$producers(s) = \{\rho \in \mathcal{N} : \rho = \langle R, M, P \rangle, s \in P\}$$

Now, for each reaction $\rho_i \in \mathcal{N}$, we generate the following constraint:

$$b_i \Rightarrow \bigwedge_{s \in R(\rho_i)} \quad \bigvee_{\rho_j \in producers(s)} b_j \tag{1}$$

Similarly, we have the following constraint that says there is a producer "on" for each modifier of ρ_i.

$$b_i \Rightarrow \bigwedge_{s \in M(\rho_i)} \quad \bigvee_{\rho_j \in producers(s)} b_j \tag{2}$$

In our running example, we have 4 reactions. Hence, we have 4 boolean variables b_1, b_2, b_3, b_4. Note that Reaction ρ_3 requires the modifier Tsc2-act, which is produced by Reaction ρ_2. Hence, we generate the constraint $b_3 \Rightarrow b_2$ of Type 2. Apart from Tsc2-act, none of the other reactants or modifiers have any producers in this network. We will discuss this issue in Section 3.3.

3.2 Competitive Inhibition

The second set of clauses encode the inhibitory effect that a reaction may have on another that shares a reactant with it.

> If a reaction is "on", then for each of its reactants, each of the other reactions that consume that reactant are "off".

For a species $s \in S$, let *consumers*(s) denote the set of reactions that have s in their set of reactants.

$$consumers(s) = \{\rho \in \mathcal{N} : \rho = \langle R, M, P \rangle, s \in R\}$$

Now, for each reaction $\rho_i \in \mathcal{N}$, we generate the following constraint:

$$b_i \Rightarrow \bigwedge_{s \in R(\rho_i)} \quad \bigwedge_{\rho_j \in (consumers(s) - \{\rho_i\})} \neg b_j \tag{3}$$

The above constraints encode the inhibitory effect one reaction has on another that shares a reactant with it. We also have similar constraints encoding competitive inhibition between reactions through a species that is a reactant in one reaction and a modifier in another.

$$b_i \Rightarrow \bigwedge_{s \in M(\rho_i)} \quad \bigwedge_{\rho_j \in (consumers(s))} \neg b_j \tag{4}$$

If two reactions share a modifier, then clearly one reaction does not inhibit the other.

In our running example, Reaction ρ_1 and Reaction ρ_2 share a common reactant, namely Tsc2. Hence, we generate the Type 3 constraints $b_1 \Rightarrow \neg b_2$ and $b_2 \Rightarrow \neg b_1$. Similarly, Reaction ρ_3 and Reaction ρ_4 compete for Rheb-act—Reaction ρ_3 uses it as a reactant, whereas Reaction ρ_4 requires it as a modifier. Hence, we generate the Type 4 constraint $b_4 \Rightarrow \neg b_3$.

3.3 Completing the Network

Biological databases of biochemical networks are often incomplete. They often use species that are not created by any reaction in the network. In the running example, Tsc2, Akt-act, Ampk-act, Rheb-act, and Mtor are all species with no producers. The presence of such species is a problem for our encoding since, to be "on", a reaction requires all of its reactants (and modifiers) to be produced by some other reaction. If there are no producers of certain species, then reactions using that species can never be turned on.

We solve this problem by adding dummy reactions that create species that have no producers. Specifically, for each species s such that $\mathit{producers}(s) = \emptyset$, we add a new reaction $\rho = \langle R, M, P \rangle$, where $R = \emptyset$, $M = \emptyset$, and $P = \{s\}$. We perform this step as a preprocessing step. As a result, these additional dummy reactions are taken into account when the constraints described above are generated.

We also encode the fact that these dummy reactions are different from other reactions by adding boolean constraints that force these reactions to be "off". For each dummy reaction ρ, if b is the corresponding boolean variable, then we add the following clause

$$\neg b \tag{5}$$

This constraint says that the dummy reaction, and hence the corresponding species, should *preferably* not be used. In Section 4, we will discuss how this preference is effected by means of weights.

In the running example, for each of the 6 species that have no producers, we add 6 new dummy reactions. Note that the addition of these 6 reactions will be reflected appropriately in the generated Type 1 and Type 2 constraints.

3.4 Optional Clauses

In case of analyzing a network instance, we may optionally have additional information about the input species, forbidden species, and target species. We now show how these are incorporated into the constraints.

Initial Species The set of species specified as initial are assumed to be present. If a set of initial species is specified, then the preprocessor adds a dummy reaction that produces all the initial species. Specifically, if S_{init} is the set of initial species, then the preprocessor will add a dummy reaction $\rho = \langle R, M, P \rangle$, where $R = M = \emptyset$ and $P = S_{init}$. Furthermore, the boolean variable b corresponding to this reaction is forced to be "on" by simply adding a clause b in the generated set of boolean constraints.

Target Species The set of target species is a list of species that should be generated by pathways generated by the tool. If a set of target species is specified, then the boolean constraint generator adds additional constraints that say that for each target species, there is at least one producer of it turned "on".

For each species s in the set of target species, we add the constraint,

$$\bigvee_{\rho_i \in producers(s)} b_i \tag{6}$$

Forbidden Species The set of forbidden species specifies the set of species that should not be used in any pathway (behavior) generated by the system. If this set is provided, then the following additional boolean constraint is generated for each species s in this forbidden set,

$$\bigwedge_{\rho_i \in producers(s)} \neg b_i \tag{7}$$

Forward Propagation The constraints that force forward firing of reactions are optional, and are generated only when the user explicitly asks for them. The forward propagation rules essentially say that for each species s, if a producer of s is "on", then some consumer of s is also "on". In other words, for each species s, the following constraint is generated,

$$\bigvee_{\rho_i \in producers(s)} b_i \Rightarrow \bigvee_{\rho_j \in consumers(s)} b_j \tag{8}$$

3.5 Mode Based Constraints

Given the above constraints, we can try to turn "on" as many reactions as possible, or turn "on" as few reactions as possible. These two possibilities are encoded as two different set of constraints.

If we wish to turn "on" as many reactions as possible (for example to find stable states), then, for each reaction $\rho_i \in \mathcal{N}$, we add the clause

$$b_i \tag{9}$$

to the set of constraints. This clause simply says that reaction ρ_i is "on".

If we wish to turn "on" as few reactions as possible (say to find minimal pathways), then, for each reaction $\rho_i \in \mathcal{N}$, we add the clause

$$\neg b_i \tag{10}$$

to the set of constraints.

4 Biochemical Pathway to Boolean Max-SAT

The constraints outlined above need not have equal weight. Some can be more important than others. This is captured by adding a weight (number) to each constraint that indicates its relative importance.

In particular, note that Type 9 constraints (and Type 10 constraints) are just hints that say reactions should be turned "on" (respectively "off"). These constraints are weaker than the other constraints that encode mutual promotion or competitive inhibition.

The choice of weights for each constraint gives additional flexibility that can be used to encode other biologically relevant information that is not generic to all biochemical processes.

In the generic translation, constraints of Type 1, Type 2, and Type 3, are each given a very large weight W. The constraint of Type 4 is given weight approximately equal to $W/2$. The constraint saying that species with no producers should not be used (Type 5 constraints) is given intermediate weight (approximately $W/(k+1)$, where k is the total number of species with no producers). Whenever present, the constraints for creation of target species and forcing forward propagation (Type 6 and Type 8 constraints) are given weight W.

The constraints that specify the hints (Type 9 and Type 10 constraints) are given weight 1.

4.1 Weighted MaxSAT

A *solution* is a mapping from the boolean variables to $\{true, false\}$. In our context, a solution maps reactions to either "on" or "off". Under a given solution, constraints also evaluate to either *true* or *false*.

Each solution can be associated with a weight: the sum of the weights of all the constraints that are made *true* by that solution. A *weighted MaxSAT solver* finds a solution that has the maximum weight.

In our running example, ignoring the dummy reactions and optional clauses for simplicity, we had generated first four clauses shown below. We now add new clauses that say each reaction should preferably be turned "on" and assign weights to each clause based on the discussion above.

c_1 :	$b_3 \Rightarrow b_2$	$w_1 = 10$	Section 3.1
c_2 :	$b_1 \Rightarrow \neg b_2$	$w_2 = 10$	Section 3.2
c_3 :	$b_2 \Rightarrow \neg b_1$	$w_3 = 10$	Section 3.2
c_4 :	$b_4 \Rightarrow \neg b_3$	$w_3 = 5$	Section 3.2
c_5 :	b_1	$w_5 = 1$	Section 3.5
c_6 :	b_2	$w_6 = 1$	Section 3.5
c_7 :	b_3	$w_7 = 1$	Section 3.5
c_8 :	b_4	$w_8 = 1$	Section 3.5

For this set of constraints, the solution $b_1 = b_4 = true, b_2 = b_3 = false$ has weight 37 (since only clauses c_6 and c_7 are violated). The weight of the solution $b_1 = b_4 = false, b_2 = b_3 = true$ is also 37 (since this solution makes clauses c_5 and c_8 false). These two are the maximum weight solutions and they correspond exactly to the two scenarios described in Section 1.

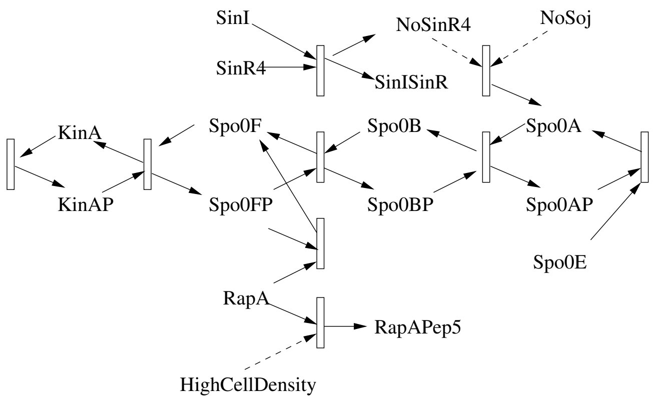

Fig. 2. Selected reactions from the sporulation initiation network of *B. Subtilis*. The reactions are represented using standard Petri net notation and show the main phosphorelay.

5 Implementation and Case Studies

We have implemented a tool based on the technique described in this paper. As a backend MaxSAT solver, we use Yices [14, 3], which is a more general *satisfiability modulo theory* solver. The input format for our tool is a network or network instance described in a very simple intermediate language. We also have several front-ends that convert from other formats to our intermediate language format. For example, we have front-ends for Pathway Logic [12, 11] and BioCyc [5, 7].

In this section, we describe the results obtained using this tool on some specific networks.

5.1 Sporulation Initiation in *B.Subtilis*

Bacillus subtilis is considered a model organism for Gram-positive bacteria and has been extensively studied in the laboratory. It is an endospore-forming bacteria most commonly found in the soil. Endospore formation is initiated when nutrients become limiting and is an adaptive response of the bacteria to their environment.

Sporulation is a one-way decision and once the decision is made, the cell undergoes changes which take 6 to 8 hours in most organisms. If conditions improve in the meantime, then the cell will be at a disadvantage. Hence the decision to initiate sporulation is important to the organism and is subject to a variety of control.

The formation of spores in Bacillus subtilis is a developmental process under genetic control. The decision to either grow vegetatively or sporulate is regulated by the state of phosphorylation of the Spo0A transcription factor [10, 6]. Spo0A

ID	Reactants	+Modifiers	⟶Products
r1		+(Spo0AP, NoSinR4)	⟶SinI
r2		+(Spo0AP, NoAbrB6, NoHpr)	⟶SinI
r3	SinI, SinR4	+	⟶SinISinR, NoSinR4
r4	SinR	+	⟶SinR4
r5		+(NoSinR4, sigmaH, NoSoj)	⟶Spo0A
r6		+(NoAbrB6)	⟶Spo0E
r7	AbrB, AbrB6	+(Spo0AP)	⟶NoAbrB6
r8		+(NoSpo0AP)	⟶AbrB
r9		+(NoAbrB6)	⟶AbrB
r10	AbrB, NoAbrB6	+	⟶AbrB6
r11	NoHpr	+(AbrB6)	⟶Hpr
r12	Hpr	+(NoAbrB6)	⟶NoHpr
r13		+(ComAP)	⟶RapA
r14	RapA	+(Spo0AP, Hpr)	⟶
r15	RapA	+(HighCellDensity)	⟶RapAPep5
r16	KinA	+(NoKipI)	⟶KinAP
r17	KinAP, Spo0F	+	⟶Spo0FP, KinA
r18	Spo0FP, RapA	+	⟶Spo0F
r19	Spo0FP, Spo0B	+	⟶Spo0BP, Spo0F
r20	Spo0A, Spo0BP	+(NoSoj)	⟶Spo0AP, Spo0B
r21	Spo0AP, Spo0E	+	⟶Spo0A, NoSpo0AP
r22		+(sigmaH, sigmaA)	⟶Spo0F
r23		+(sigmaA)	⟶Spo0B
r24	KipI	+(NoFood, NoNitrogen)	⟶NoKipI

Table 1. The list of reactions modeling the sporulation initiation network.

obtains its phosphate through a phosphorylation pathway (see Figure 2), the so-called *phosphorelay*, in which at least three histidine protein kinases transfer phosphate to the relay protein, Spo0F, then to Spo0B, and finally to Spo0A (represented by ReactionIDs $r17$, $r19$, and $r20$ in Table 1). In addition, the phosphorylation state of Spo0A is modulated by specific phosphatases, such as Spo0E, which dephosphorylates Spo0A-P, and RapA, which dephosphorylates Spo0F-P (ReactionIDs $r18$, $r21$).

The *SinI* and *SinR* pair is a regulatory operon in the sporulation initiation network. While SinR is a transcriptional regulator that represses *spo0A* transcription, SinI disrupts the SinR tetramer through the formation of a SinI-SinR heterodimer. This aspect, along with the logic regulating SinI transcription, is encoded in ReactionIDs $r1$, $r2$, $r3$, and $r4$.

The activity of protein RapA is modulated by quorum sensing, the process of sensing activity in neighboring cells and reacting in a cell-density-specific fashion. Under high population density, RapA is inhibited by PhrA pentapeptide (not modeled in the reactions). These aspects are captured in ReactionIDs $r13$, $r15$. The protein kinase KinA is a sensor that initiates the phosphorelay and is modeled here by ReactionIDs $r16$, $r17$. Most of the remaining reactions encode transcriptional regulation logic for different proteins.

On this simplified model of sporulation initiation, the tool implementing the approach described in this paper can find possible stable behaviors of the network. These behaviors are found as subsets of reactions in the network that can be consistently "on". The tool finds 3 different possibilities for the model above.

- SinI is produced, and it binds to SinR, thus preventing it from repressing *spo0A*. RapA is converted to RapAPep5, thus preventing it from dephosphorylating Spo0A-P. In the presence of stress signals, KipI is prevented from inhibiting KinA from self-kinasing. The self-kinasing of KinA triggers the phosphorelay, which leads to production of Spo0A-P, a precursor for sporulation.
- In the second stable state scenario, RapA dephosphorylates Spo0F-P, thus breaking the phosphorelay chain. Thus, there is no production of Spo0A-P.
- The third stable state scenario is similar to the first, except that Spo0E dephosphorylates the produced Spo0A-P, thus using up the produced Spo0A-P.

The three stable scenarios each make different assumptions about the environment. In our case, the environment consists of the species that are not created by any of the reactions in the network. In the network above, HighCellDensity, and NoFood, are two examples of *input species*.

The tool can also be used in the mode in which a desired target set of species is specified (for example, Spo0A-P). In this case, the tool will generate the first stable scenario above to show how Spo0A-P could be produced.

5.2 MAPK Signaling Network

The Mitogen-Activated Protein kinase (MAPK) network regulates several cellular processes, including the cell cycle machinery. The MAPK cascade communicates signals from growth factors that bind receptor kinases to transcription and other cellular processes [2]. A simplified model of this network, taken from [2], can be encoded in our notation as shown in Table 2. The tool finds two stable sets of behavior for this network.

- The positive feedback loop is active. In this case, either Grb2, Sos1, or PKC* turns on Ras. This causes, in steps, the phosphorylation of Raf, MEK, and Erk. Activated Erk causes production of AA*, which stimulates PKC.
- The negative feedback loops are active. In this case, protein phosphatase 2A (PP2A) dephosphorylates both Raf* and Mek*, and MKP dephosphorylates Erk*. MKP is created by transcription of *MKP* gene, and this is promoted by Erk*.

The two stable solutions clearly identify the positive cycle and the multiple negative cycles that break the positive cycle. The overall system behavior is

ID	Reactants+Modifiers		⟶Products
r1	Ras	+(Grb2, Sos1)	⟶Ras*
r2	Ras	+(PKC*)	⟶Ras*
r3	Raf	+(Ras*)	⟶Raf*
r4	Raf*	+(PP2A)	⟶Raf
r5	Mek	+(Raf*)	⟶Mek*
r6	Mek*	+(PP2A)	⟶Mek
r7	Erk	+(Mek*)	⟶Erk*
r8	Erk*	+(MKP)	⟶Erk
r9		+(Erk*, MKPgene)	⟶MKP
r10	AA	+(Erk*, Ca)	⟶AA*
r11	PKC	+(DAG, Ca, AA*)	⟶PKC*

Table 2. The list of reactions modeling the MAPK signaling network.

seen to be a result of the close interaction between the positive and negative cycles.

We also used the detailed model of the MAPK signaling network from [1]. The total running time on the full network is of the order of a few seconds.

5.3 EGF Stimulation Network

In the Pathway Logic project [12, 11], a model of Egf stimulation is being developed by curating a network of biochemical reactions involved in mammalian cell signaling from the literature. When a cell is stimulated by Egf, certain species are experimentally observed to be present in the cell after its initial stimulation. These observations can be used to validate the model by checking whether the model predicts the observations. To carry out the validation, we started with a network of about 400 reactions and created a network instance by adding initial and target species. Specifically, we started with a set of about 250 *initial* species and 62 *target* species that are experimentally observed in response to EGF stimulation.

When this network instance is analyzed by our tool, our tool attempts to find a set of reactions that will create each of the target species using the initial species and the reactions in the network. A "–no-assume" option tells the tool to not assume any species not already specified in the initial set. (Recall that, by default, species that have no producers can be assumed, with a moderate penalty.)

The output of the tool indicated that it was not possible to find a solution without violating one Type 3 and one Type 4 *competitive inhibition* constraints. Specifically, the species (Frap1:Lst8)-CLc [2] is a reactant in two different reactions that are *both* required to be "on" to create the target species. This causes a Type 3 constraint to be violated. The Type 4 constraint that is violated is caused by the species Src-CLi, which is used as a reactant in a reaction to create

[2] a complex containing Frap1 and Lst8 located in the cytoplasm,CLc

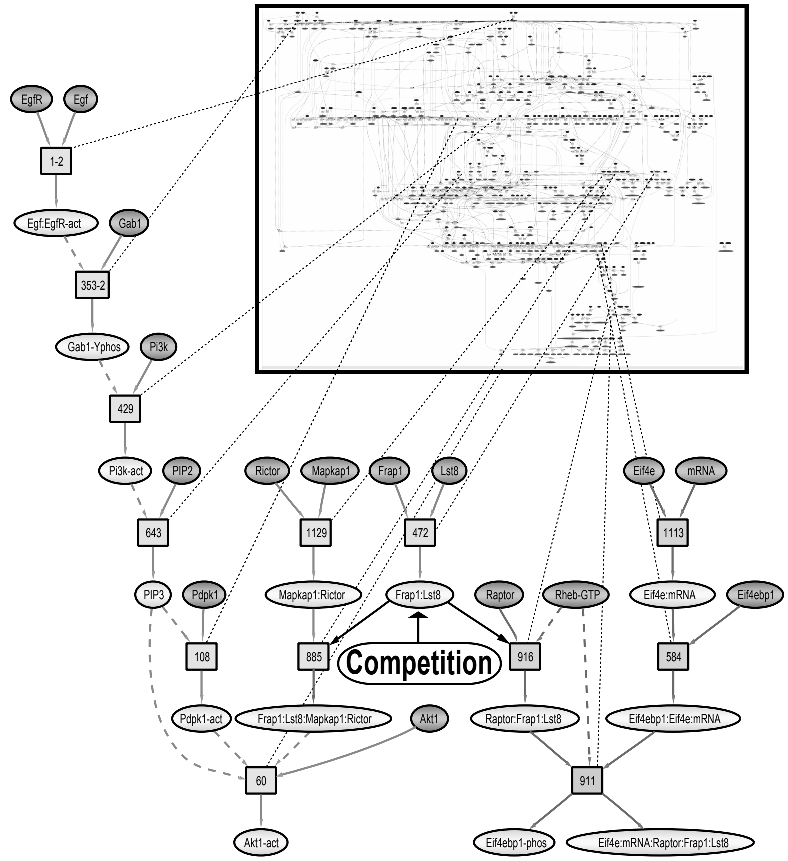

Fig. 3. A simple network with competing rules

Src-act-CLi, and it is also used as a modifier in the reaction that creates Cbl-Yphos-CLi. This violation pointed out a typing error in specifying the reaction rules which has been corrected. Figure 3 shows the pathways competing for (Frap1:Lst8)-CLc in the context of the larger network.

Using our tool provided two valuable forms of feedback to the model developer. One was a form of meta analysis or type-checking that detected syntactic problems with the model. (The first pass detected a number of inconsistencies that were easily repaired.) The second was the identification of the point of competition. Using the Pathway Logic Assistant [13] one can check whether a given set of observations is predicted, singly or jointly. However if a prediction fails there is no feedback as to the cause of failure. Using MaxSAT, candidate conflicting constraints can be identified to guide the modeler.

Starting with the discovered Type 3 violation and studying the subnetwork connected to this reaction lead to two hypotheses: (1) (Frap1:Lst8)-CLc splits into two *populations* one for each of the two competing reactions; (2) there is a feedback loop that can reset the state of (Frap1:Lst8)-CLc and the system oscillates between the two pathways. Experiments are ongoing to test these hypotheses.

6 Related Work

We compare here with work that is closer in spirit to our work, and do not mention all the literature devoted to building various kinds of models and improving understanding of specific biological phenomena, such as sporulation and MAPK signaling.

Senachak et. al. [8] give a generic interpretation to a reaction network by translating it to a graph. Strongly-connected components of the graph are related to the pathways. The construction of the graph has some unusual steps, such as *cascading*, that arise primarily because the authors use species as defining the nodes of the graph. The main difference in our approach is that, in our approach, the boolean variables correspond to reactions in the network. We believe this leads to a much simpler and natural encoding of the “cascading”-style constraints of [8].

7 Conclusion

We presented a new approach for analyzing biochemical reaction networks using MaxSAT. The novelty here is that we make reactions central to the notion of a steady-state behavior. A steady-state behavior is a subset of reactions that can be mutually consistently “on”.

The attractiveness of our approach is that it is generic and applies to networks coming from different kinds of biological networks. Additionally, it is also flexible and allows encoding of knowledge specific to certain kinds of networks via suitable manipulation of the weights on the generic constraints.

The analysis approach is promising. Even for the largest networks we have studied, the analysis takes at most a few seconds to compute answers.

Possible future work include studying quantitative variants of the boolean constraints. Fortunately, our backend tool, Yices, supports reasoning over linear arithmetic constraints. We can replace the use of boolean MaxSAT with MaxSAT over arbitrary combination of boolean and linear arithmetic constraints.

References

1. U. S. Bhalla and R. Iyengar. Robustness of the bistable behavior of a biological signalling feedback loop. *Chaos*, 11(1), 2001.

2. U. S. Bhalla, P. T. Ram, and R. Iyengar. MAP kinase phosphatase as a locus of flexibility in a Mitogen-Activated Protein kinase signaling network. *Science*, 297, 2002.
3. B. Dutertre and L. M. de Moura. A fast linear-arithmetic solver for dpll(t). In *Computer Aided Verification, 18th International Conference*, volume 4144 of *LNCS*, pages 81–94. Springer, 2006.
4. F. Fages, S. Soliman, and N. Chabrier-Rivier. Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. *Journal of Biological Physics and Chemistry*, 4(2):64–73, 2004.
5. I.M. Keseler, J. Collado-Vides, S. Gama-Castro, J. Ingraham, S. Paley, I.T. Paulsen, M. Peralta-Gil, and P.D. Karp. EcoCyc: A comprehensive database resource for Escherichia coli. *Nucleic Acids Research*, 33:D334–7, 2005.
6. L. M. Prescott, D. A. Klein, and J. P. Harley. *Microbiology.* McGraw-Hill, 2002.
7. P. Romero, J. Wagg, M.L. Green, D. Kaiser, M. Krummenacker, and P.D. Karp. Computational prediction of human metabolic pathways from the complete human genome. *Genome Biology*, 6(R2):1–17, 2004.
8. J. Senachak, M. Vestergaard, and R. Vestergaard. Rewriting game theory and protein signalling in MAPK cascades. In *Proc. CMSB*, 2006.
9. Carron Shankland, Nam Tran, Chitta Baral, and Walter Kolch. Reasoning about the ERK signal transduction pathway using BioSigNet-RR. In G. Plotkin, editor, *Proceedings of the Third International Conference on Computational Methods in System Biology*, 2005.
10. P. Stragier and R. Losick. Molecular genetics of sporulation in bacillus subtilis. *Annu. Rev. Genet.*, 30:297–341, 1996.
11. C. Talcott, S. Eker, M. Knapp, P. Lincoln, and K. Laderoute. Pathway logic modeling of protein functional domains in signal transduction. In *Proceedings of the Pacific Symposium on Biocomputing*, January 2004.
12. Carolyn Talcott. Symbolic modeling of signal transduction in pathway logic. In L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, editors, *2006 Winter Simulation Conference*, 2006.
13. Carolyn Talcott and David L. Dill. Multiple representations of biological processes. *Transactions on Computational Systems Biology VI*, 4220:221–245, 2006.
14. http://yices.csl.sri.com/. Yices home page.