Reachability Analysis based Model Validation in Systems Biology

Yang YANG
Department of Electrical & Computer Engineering
National University of Singapore
Singapore
yang82@nus.edu.sg

Hai LIN
Department of Electrical & Computer Engineering
National University of Singapore
Singapore
elelh@nus.edu.sg

***Abstract*—Systems biology is an emerging multi-disciplinary area, which aims to understand the underneath regulatory mechanisms of the biomolecular interaction networks inside the cell through dynamical system approaches. The first challenge in systems biology is how to obtain an accurate and predictable computational model for the biomolecular networks under study. However, due to limited experimental data, it is unavoidable to have incomplete or even wrong models. Therefore, it is a critical task in systems biology to check the model's correctness, which is called model validation problem. This paper will focus on this issue, and propose a (un-)reachability analysis based model validation method. In particular, Petri net models are investigated, and the validation process is evaluated by the reachability of state equations. It is shown that the reachability can be checked by the existence of integer solutions of Diophantine equations. Two methods are proposed to solve the equations. The first one is by Smith normal form test, and the other is by integer programming. Two case studies are provided to demonstrate these two approaches. These tests can screen out the unreachable states and offer the hints to modify the model structure, which provides us more insights of the regulatory mechanism and helps biologists to generate hypotheses and design experiments.**

***Index Terms*—Systems Biology, Petri Nets, Reachability Analysis**

I. INTRODUCTION

After DNA was deciphered, biologists are trying to make clear that life is chemistry and physics and believe that once we have found the smallest components of life, we will be able to understand the whole. However, the current limitations of component-based research have brought many problems which lead to a revival of holistic approaches [1]. Meanwhile, new developments in post-omics technology provide researchers with vast data sources to study the regulatory networks in a systematic fashion. There is a high demand of integrated and comprehensive investigations of biological systems. The knowledge gained from these analyses provides valuable insights which help in generating hypotheses and designing experiments. This whole interactive cycle will shed more light on understanding the biological systems. To achieve this goal, it requires novel methods from inter-disciplinary fields, which are completely different from traditional trial-and-error styles, to model and analyze the biological systems [2].

Major part of the study in systems biology is carried out through mathematical/computational models of the biomolecular networks. Many formalisms can be the candidates to describe the inherent mechanisms, such as ordinary and partial differential equations in the continuous domain, Boolean network and Petri nets in the discrete domain, Bayesian network in the stochastic domain, and so on [3] [4]. On the other hand, due to limited experimental data, it is unavoidable to have incomplete or even wrong models. Therefore, it is a critical task in systems biology to check the model's correctness, which is called model validation problem. For example, in the genetic regulatory network, incomplete information of relations will lead to uncertainties of connections among genes and regulatory proteins. Sometimes, the model needs to tolerate the inconsistent information from different sets of experimental data. Under such circumstances, the models are required to be step-wisely developed and validated to increase the confidence to reflect the reality.

In this paper, we present a new idea which facilitates the validation of reachable biological states and the model structure. We adopt the model built in Petri nets. Then, instead of reachability graph and model checking, we test the reachability property from the basic state equations and the existence of integer solutions of Diophantine equations. Furthermore, we provide two easily executable methods to do the verifications. We illustrate our approaches by analyzing two examples, biochemical reactions and lac operon genetic expression pathway in *E.coli*.

This paper is organized as follows. In Section II, we give an essential background of Petri nets. In Section III, we derive the reachability criterion through Smith normal form and formulate the problems in the form of integer programming. In Section IV, we consider two case studies to apply our techniques to carry out the validations. Finally, in Section V, we end this paper with concluding remarks.

II. BACKGROUND AND PROBLEM FORMULATION

A. Petri nets

Petri nets have a graphical representation. Simple but pragmatic interface makes Petri nets user-friendly and popular. A Petri net is a directed, bipartite multi-graph. Bipartite nodes consist of *places* and *transitions*. Graphically, places are denoted as circles, and transitions are rectangles or bars. Arcs are used to connect places with transitions. Same types of nodes have no arcs linking in between.

978-1-4244-6502-6/10/$26.00

The meanings of the transitions and places depend on the modeling context. For example, in event/condition systems, a transition stands for the occurrence of an event, while the input places of the transition are the pre-conditions, all the necessary conditions needed to trigger the event; and the output places are post-conditions, all the conditions satisfied after the occurrence of the event. In biochemical reactions, transitions represent chemical reactions. The input places and output places are respectively the reactant compound and product compound in these reactions. Sometimes, one net can have mixed types of places and transitions for different interpretations.

Petri nets are executable. The execution is based on the idea of *tokens*, which are graphically depicted by dots and reside in places. The distribution of tokens in places is called *marking*. Firing of transitions will change the distribution of tokens. Hence, Petri net is also dubbed "token game". The marking of all places comprises the state space of Petri net. One example is given in Fig. 1.

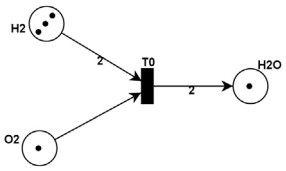

Fig. 1. Petri net example showing the synthesis of water molecule. There are 3 places, H2, O2 and H2O, 1 transition T_0. The weights of arc connecting from place H2 to transition T_0 and T_0 to place H2O are 2. The weight of arc connecting from O2 to T_0 is one, usually omitted by default.

The definitions of Petri nets are mathematically expressed in the following forms, which are adapted from [5].

Definition 1: A Petri net is a 5-tuple, $PN = (P,T,F,W,M_0)$ where,

$P = \{p_1, p_2, \cdots, p_m\}$ is a finite set of m places,
$T = \{t_1, t_2, \cdots, t_n\}$ is a finite set of n transitions,
$F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs,
$W : F \to \{1,2,3,\cdots\}$ is a weight function,
$M_0 : P \to \{0,1,2,3,\cdots\}$ is the initial marking,
$P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.

The weight of an arc from place p_i to transition t_j, and from transition t_j to place p_i is represented as $w(p_i,t_j)$ and $w(t_j,p_i)$, respectively. The pre- and post-set of transitions $t \in T$ are $\bullet t = \{p \mid w(p,t) > 0\}$ and $t\bullet = \{p \mid w(t,p) > 0\}$, respectively. Similarly, the pre- and post-set of places $p \in P$ are $\bullet p = \{t \mid w(t,p) > 0\}$ and $p\bullet = \{t \mid w(p,t) > 0\}$, respectively. The state of a Petri net is the marking $M : P \to \{0,1,2,\cdots\}$, a nonnegative integer-valued vector. The i-th component of the marking vector, denoted by $M(p_i)$, represents the number of tokens at place p_i.

Definition 2: A transition is said to be *enabled* in marking M, if all places in its pre-set have at least as many tokens as the weights of the arcs connecting those places to the transition, that is, if $\forall p \in \bullet t, M(p) \geq w(p,t)$, transition t is said to be *enabled* in the marking M.

Definition 3: Firing an enabled transition changes the token distribution by removing as many tokens as the weights on the arcs leading in from all its pre-set places and adding as many tokens as the weights on the arcs leading out to all its post-set places, that is, by firing an enabled transition t, the change in token distribution from M to M' is given by

$$M'(p) = \begin{cases} M(p) - w(p,t) & \forall p \in \bullet t \\ M(p) + w(t,p) & \forall p \in t\bullet \\ M(p) & \forall p \notin \{\bullet t \cup t\bullet\} \end{cases}$$

In the example shown in Fig. 1, the transition T_0 is enabled, and firing the transition makes the marking change from M_0(H2,O2,H2O) $= [3\ 1\ 1]^T$ to M_1(H2,O2,H2O) $= [1\ 0\ 3]^T$.

These changes in the state of the system can be represented in an algebraic form using the state equations given by

$$M_k = M_{k-1} + C^T \mathbf{u}_k,$$

where $C = [c_{ij}]$ is an $n \times m$ incidence matrix of integers; the superscript T stands for transpose; $\mathbf{u}_k$ is called firing or control vector, which is an $n \times 1$ column vector of $n-1$ **0**'s and one nonzero entry, a **1** in the i-th position indicating that transition i fires at the k-th firing. The typical entry of incidence matrix C is given by

$$c_{ij} = c_{ij}^+ - c_{ij}^-,$$

where $c_{ij}^+ = w(i,j)$ is the weight of the arc from transition i to its output place j and $c_{ij}^- = w(j,i)$ is the weight of the arc to transition i from its input place j.

Theorem 1: (Necessary Reachability Condition)[5] Suppose that a destination marking M_d is reachable from initial marking M_0 through a firing sequence $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_d\}$. The state equations can be derived as follows.

$$M_d = M_0 + C^T \sum_{k=1}^{d} \mathbf{u}_k, \tag{1}$$

which can be rewritten as

$$C^T \mathbf{x} = \triangle M, \tag{2}$$

where $\triangle M = M_d - M_0$ and $\mathbf{x} = \sum_{k=1}^{d} \mathbf{u}_k$.

Here $\mathbf{x}$ is an $n \times 1$ column vector of nonnegative integers and is called the firing count vector. The i-th entry of $\mathbf{x}$ denotes the number of times that transition i must fire to transform M_0 to M_d.

By Theorem 1, the existence of a nonnegative integer-valued solution of state equations (2) is a necessary condition for the reachability of Petri net, i.e. if the destination marking M_d is reachable from the initial marking M_0, there exists nonnegative integer solution for state equations (2). The sufficiency of this condition is only restricted to acyclic Petri nets [6].

B. Problem Formulation

The above necessary condition could be treated into the following two scenarios, which will give hints to do the model validation in systems biology.

First of all, the contrapositive of the necessary condition provides a sufficient condition for unreachability, i.e. if the state equations do not have nonnegative integer solutions, M_d is not reachable from M_0. This condition can be used to test

the unreachability of certain states. Furthermore, in the acyclic case, this condition is used to test both the reachability and unreachability of concerned states at the same time.

In the biological context, taking the genetic regulatory pathways for example, this condition could be utilized to test the expressions of certain genes. Initial marking could be generated according to the initial expression information. The incidence matrix could also be constructed when the relations among the genes are identified. Then, this condition may help to verify the biologists' hypotheses on the interested gene's expression. It can be imagined that we may often eliminate some impossible guesses through the state equations, which will save remarkable cost and energy for the experimenters.

Secondly, if the expression data are observed, which means, in Petri net formalism, the state equations should be definitely solvable. If it has nonnegative integer solutions accordingly, the model of current Petri net is validated. It can be further analyzed to get more insights and generate further hypotheses based on the current model. If there does not exist a non-negative integer solution, the incidence matrix C is somehow wrong. We have to modify the structure of the Petri net, such as the connections and weights, reflecting the reactions and stoichiometric parameters.

To address this problem, we introduce the following two approaches. One is the Smith normal form based criterion, which is established to check the reachability of desired states. It is shown that if the condition violates the criterion, the state equations are unsolvable. Accordingly, the desired state cannot be reached. The other approach is to test the model structure by the solvability of state equations under different initial conditions. The model which agrees with known evidences will be retained. Integer programming is introduced to facilitate this problem.

III. REACHABILITY CRITERION

A. Diophantine Equations

The discussion in the last section transforms the reachability problem to the existence of nonnegative integer solutions of state equations (2). By inspection of the state equations, the entries of the incidence matrix are all integers. Additionally, the solutions are also required to be integers, which is correlated to the Diophantine Analysis.

In mathematics, a Diophantine equation is an indeterminate polynomial equation that allows the variables to be integers only. The solvability of Diophantine equations is proposed as the tenth of Hilbert problem. In general, Diophantine problems are unsolvable. But in the case of linear Diophantine equation, Bézout's identity is a perfect solution [7]. Bézout's identity is specific to one linear Diophantine equation with multiple unknowns. In the sequel, we will analyze a set of Diophantine equations using linear algebraic method first. Then, this problem will be reformulated as integer programming (IP), and the state equations are solved numerically.

B. Smith Normal Form Test

The Smith normal form applies for any matrix with entries in a principal ideal domain (PID). In particular, the integers are PID, so one can always calculate the Smith normal form for an integer matrix [8].

Theorem 2: [9] If $A=(a_{ij})$ is any integer matrix of order $m\times n$ with rank r, then there exists a unique integer matrix

$$D_{m\times n}=\begin{bmatrix} d_1 & 0 & 0 & \cdots & 0 & & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 & & \cdots & 0 \\ 0 & 0 & \ddots & & \vdots & & & \vdots \\ \vdots & \vdots & & & d_r & 0 & \cdots & 0 \\ 0 & 0 & & & 0 & 0 & & \vdots \\ \vdots & \vdots & & & & \ddots & \ddots & \\ 0 & 0 & \cdots & & \cdots & & \cdots & 0 \end{bmatrix},$$

where $d_k>0, k=1,\cdots,r$, and d_k divides d_{k+1}. D is the Smith normal form of A, and $\{d_1,d_2,\cdots,d_r\}$ are called invariant factors. Furthermore, the matrices A and D are equivalent ($A\cong D$), i.e. $A=PDQ$ for some unimodular square matrices P of order m and Q of order n.

The Smith normal form D can always be obtained through a series of elementary row and column operations on the original matrix A by left-multiplying matrix P and right-multiplying matrix Q correspondingly. Several software packages are also available for this computation, such as Fermat, GAP, MAGMA and so on.

The classical method using the Smith normal form to solve the linear Diophantine equations is described in [9]. In the following, one simple criterion to test the the existence of integer solution is presented. Before that, a lemma is given first.

Lemma 1: There exist matrix $A_{m\times n}=(a_{ij})$ and vector $\mathbf{b}=[b_1,b_2,\cdots,b_m]^T$. Two matrices $A_0=[A\ \mathbf{0}]_{m\times(n+1)}$ and $\overline{A}=[A\ \mathbf{b}]_{m\times(n+1)}$ are equivalent if and only if there exists an unimodular matrix C with order $n+1$, such that $\overline{A}=A_0C$.

Proof: The sufficiency is obvious. The necessity can be derived by the property that elementary row and column operations do not change the rank of matrix. And the greatest common divisor of every k-th subdeterminant of two equivalent matrices is the same. ■

Theorem 3: Linear equations $AX=\mathbf{b}$, where $A,X,\mathbf{b}$ are integer matrices of size $m\times n$, $n\times 1$ and $m\times 1$, respectively, have integer solutions if and only if matrices A and $\overline{A}=[A\ \mathbf{b}]$ share the same set of invariant factors of the Smith normal form.

Proof: Necessity: Suppose $AX=\mathbf{b}$ has integer solutions $\mathbf{x}=[x_1,x_2,\cdots,x_n]^T$. Multiply the first n columns of $\overline{A}$ by $-x_1,-x_2,\cdots,-x_n$, respectively, and add the products to the $(n+1)$th column. Then, A_0 is obtained. Consequently, A and $\overline{A}$ share the same set of invariant factors.

Sufficiency: If A and $\overline{A}$ have the same invariant factors, so is for A_0 and $\overline{A}$. Therefore, $\overline{A}$ is equivalent to A_0. According

to Lemma 1, there exists an unimodular matrix C with order $n+1$, such that $\overline{A} = A_0 C$.

Suppose the last column of C is $C_{n+1} = [c_1, c_2, \cdots, c_n, c_{n+1}]^T$. The equations $AC_{n+1} = \mathbf{b}$ hold. Equivalently, the equations $AX = \mathbf{b}$ have integer solutions. ■

By Theorem 3, it can be verified whether the given state equations of Petri net have integer solutions. This theorem can be efficiently applied by taking advantages of computational software. One limitation here is that the theorem only guarantees the existence of integer solutions. Nonnegativity cannot be addressed by this theorem. More efforts are needed to completely solve this problem. However, it does provide an efficient way to do the first test. If this test cannot pass, the candidate will be definitely eliminated in the first round.

C. Integer Programming Approach

Compared to the first approach, a more straightforward method is to solve the equations numerically. Because one main concern in practice is to confine the whole experimental procedure to be as simple and quick as possible, this concern can be formulated as an integer programming problem [10]. The total number of reactions or efforts can be the objective function to be minimized, and the constrains are defined by the state equations. One example is shown as follows.

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} x_i \\ \text{subject to} & C^T \mathbf{x} = \triangle M \\ & x \in Z, x \geq 0 \end{array}$$

In addition, in genetic regulatory networks, the markings of places usually take binary values, i.e. 0 or 1. For instance, one gene is expressed high, which can be represented with "1" token in the corresponding place. Otherwise, it is represented with "0" token. On the other hand, the transitions usually represent these occurrences of reactions. Sometimes, to test the gene expression in terms of logical relations, some transitions are required to happen at most once, implying that the transforms are enabled or disabled, as the reasoning process does. The above considerations can be categorized as 0-1 integer programming or binary integer programming (BIP) where variables are required to be only 0 or 1. Several supporting software can solve BIP problems. Among them, LINDO is a powerful optimization tool for linear programming, integer programming, etc [11]. We may use the built-in numerical algorithm to solve the state equations. The solvability and nonnegativity can be tested directly from the solutions.

IV. Case Studies

In this part, two case studies will be presented to show applications of the aforementioned methods. In the first example, one biochemical reaction network is built in Petri net. Using Smith normal form test, one expected state is proven to be unreachable. The second example is derived from the lac operon genetic regulatory network in *E.coli*. The example will demonstrate the modification of model structure by binary integer programming.

A. Biochemical reactions

Suppose that there exist three molecules P1, P2 and P3. Three reactions can happen among them. The detailed relations are illustrated below. The enzyme molecules are omitted in the corresponding reactions.

Reaction 1: $P_2 \rightarrow P_1 + 4P_3$
Reaction 2: $2P_3 + P_2 \rightarrow 2P_1$
Reaction 3: $3P_1 \rightarrow P_2$

The Petri net model is shown in Fig. 2 .

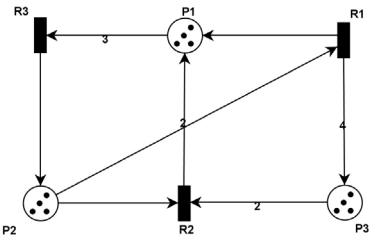

Fig. 2. Petri net representation of case study 1. The numbers on the arcs denote the weights. Weights of 1 are omitted. Initial marking M_0 is $[4\ 4\ 4]^T$.

The model contains 3 places and 3 transitions. Therefore, it has a 3×3 incidence matrix.

$$C_1 = \begin{bmatrix} 1 & -1 & 4 \\ 2 & -1 & -2 \\ -3 & 1 & 0 \end{bmatrix}.$$

The state equations can be written as,

$$C_1^T x = \begin{bmatrix} 1 & 2 & -3 \\ -1 & -1 & 1 \\ 4 & -2 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \triangle M$$

The initial marking M_0 is $[4\ 4\ 4]^T$. The tokens can represent the numbers of molecules or scaled concentrations. One expected result M_{d1} is $[5\ 1\ 6]^T$.

$$\triangle M_1 = M_{d1} - M_0 = \begin{bmatrix} 5 \\ 1 \\ 6 \end{bmatrix} - \begin{bmatrix} 4 \\ 4 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \\ -3 \\ 2 \end{bmatrix}$$

Next, we will perform the Smith test to check whether this expected state can be reached or not.

Smith normal form of C_1^T is

$$C_1^T \cong SNF1 = diag\{1, 1, 8\};$$

while, Smith normal form of $[C_1^T \ \triangle M_1]$ is

$$[C_1^T \ \triangle M_1] \cong SNF2 = [diag\{1, 1, 2\}, \mathbf{0}_{3\times 1}].$$

The invariant factors are different. According to Theorem 3, these state equations cannot have integer solutions. It can be concluded that the desired state of three molecules cannot be realized. This conclusion can also be verified manually.

If the destination state is $M_{d2} = [6\ 2\ 4]^T$. Then, $\triangle M_2 = [2\ -2\ 0]^T$. Following the same procedure as above, the Smith normal form of new augmented matrix $[C_1^T \ \triangle M_2]$ is

$$[C_1^T \ \triangle M_2] \cong SNF3 = [diag\{1, 1, 8\}, \mathbf{0}_{3\times 1}].$$

The same set of invariant factors guarantees the existence of integer solutions. And the solutions are $[x_1\ x_2\ x_3]^T = [1\ 2\ 1]^T$, which represent after firing R1 and R3 once, and firing R2 twice, the desired state can be obtained under the current initial marking. However, the solutions of state equations cannot determine the firing sequences. This drawback can be overcome by reachability graph complementarily provided that the token distribution allows so.

B. lac Operon pathway

This section demonstrates how we can use the integer programming to solve the state equations, modify the undesired behavior generated by the model and correct the model structure accordingly. Biological facts used for constructing the model are from the well-studied lac operon gene regulatory network happening in *E.coli* [12].

During the gene's transcription, the activity of a single promoter can be controlled by two different signals. The lac operon in *E.coli*, for example, is controlled by both the lac repressor and the activator protein CAP. Glucose and lactose levels control the initiation of the lac operon through their effects on the lac repressor protein and CAP. Addition of lactose increases the concentration of allolactose which binds to the repressor protein and removes it from the DNA. Addition of glucose decreases the concentration of cyclic AMP (cAMP), then the dissociation of cAMP with CAP removes the activator CAP from the DNA, which will lead to the shutdown of transcription. The transcripted gene of lacZ, lacY and lacA will also join this regulatory network. In this work, we only consider the initiation of the transcription under control of lactose and glucose.

It has been confirmed that, among four combinations of lactose and glucose, only the first condition illustrated in Table I will turn on the transcription.

TABLE I
TRANSCRIPTION ON/OFF BY COMBINATIONS OF LACTOSE AND GLUCOSE

condition	lactose	glucose	transcription
1	+	-	yes
2	+	+	no
3	-	-	no
4	-	+	no

Based on the description above, the relations among components are established. The first model is shown in Fig. 3. Most places are the involved proteins, such as lactose, glucose, etc. However, place P_7 is used to represent the gene's status "transcription start". Transition T_1 indicates that the existence of lactose will lead to the generation of allolactose. T_6 has the similar purpose. The transitions, $T_2, T_3, T_4, T_5, T_7, T_9$, which have only input arcs or output arcs are called sink transition or source transition, respectively. As the name implies, source transition can only produce tokens. For example, T_3, T_4, T_5 will produce the repressor protein, CAP and cAMP, respectively. On the other hand, sink transition will only consume tokens. For example, glucose will consume cAMP by T_7. T_2 represents that allolactose will make the repressor malfunction. T_9 is denoting the suppression by "repressor". Once T_9 fires, it will consume "complex" on the existence of "repressor". Thus, T_6 will be disabled, and transcription cannot start.

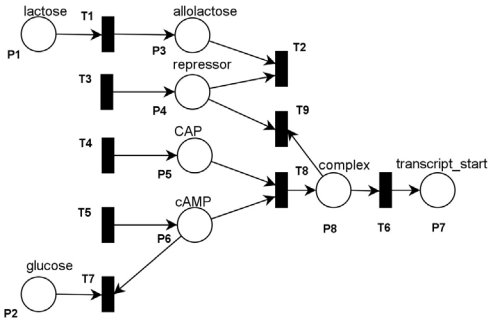

Fig. 3. Petri net representation of Case study 2 Model 1.

As explained earlier, this problem can be formulated as binary integer programming. The model should be tested by the four combinations in Table I, which correspond to four initial markings $M_{10}(P_1, P_2, \cdots) = [1\ 0\ \mathbf{0}_6]^T$, $M_{20}(P_1, P_2, \cdots) = [1\ 1\ \mathbf{0}_6]^T$, $M_{30}(P_1, P_2, \cdots) = [0\ 0\ \mathbf{0}_6]^T$, $M_{40}(P_1, P_2, \cdots) = [0\ 1\ \mathbf{0}_6]^T$. The final marking is $M_F(P_1, P_2, \cdots, P_7, P_8) = [\mathbf{0}_6\ 1\ 0]^T$, in which the entry "1" at P_7 position denotes the initiation of transcription. Thus, there will be four state equations. If the state equations are expressed as $A^T\mathbf{x} = \mathbf{b}$, then there will be four candidates of $\mathbf{b}$. The incidence matrix of A is as follows.

$$A_{9\times 8} = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Four candidates of b are

$$\begin{aligned} \mathbf{b}_1 &= [\ -1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ ]^T, \\ \mathbf{b}_2 &= [\ -1\ -1\ 0\ 0\ 0\ 0\ 1\ 0\ ]^T, \\ \mathbf{b}_3 &= [\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ ]^T, \\ \mathbf{b}_4 &= [\ 0\ -1\ 0\ 0\ 0\ 0\ 1\ 0\ ]^T. \end{aligned}$$

By Smith normal form test, all four conditions have same invariant factors set, implying that all four state equations have integer solutions. However, it is required that all the values of markings are binary value, i.e. "0" or "1". Therefore, Smith test cannot help further. Binary integer programming is used

to solve this problem. The formulation is listed below.

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{9} x_i \\ \text{subject to} & x_i = \{0,\ 1\} \\ -x_1 & = \alpha \\ -x_7 & = \beta \\ x_1 - x_2 & = 0 \\ -x_2 + x_3 - x_9 & = 0 \\ x_4 - x_8 & = 0 \\ x_5 - x_7 - x_8 & = 0 \\ x_6 & = 1 \\ -x_6 + x_8 - x_9 & = 0 \end{array}$$

where $\{\alpha, \beta\}$ can be $\{-1,0\}$, $\{-1,-1\}$, $\{0,0\}$ or $\{0,-1\}$.

Using LINDO, condition 2 and 4 cannot find the solutions subject to binary values, meaning that the initial markings under these two conditions cannot initiate the transcription, which accords with the evidence. Condition 1 can find the solution, which also coincides with the evidence that presence of lactose and absence of glucose initiate the transcription. One unexpected result is case 3. It is also solvable subject to binary value, and the solution is $\mathbf{x} = [\ 0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 0\ \ 1\ \ 0\ ]^T$.

Firing T_4, T_5, T_6, T_8 can also make the desired state reachable. This result contradicts the evidence, and provides the warning of incorrectness of the model structure. After check, the model cannot guarantee whenever "repressor" contains 1 token, it has to be consumed by transition T_9. So the model has to be modified accordingly.

In the new model shown in Fig. 4, one intermediate place P_9, "rep_bar" is added to indicate the status that the repressor is consumed by allolactose. This place collaborates with "complex" to enable T_6 and finally initiates the transcription.

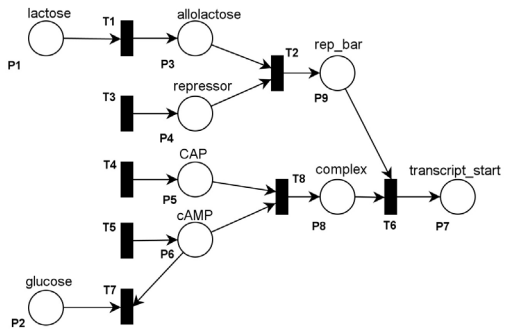

Fig. 4. Petri net representation of Case study 2 Model 2.

The incidence matrix becomes a 8×9 matrix with the following form.

$$A_{8\times 9} = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 & 0 \end{bmatrix}$$

The initial and final markings are similar with the previous model. The binary integer programming problem becomes

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{8} x_i \\ \text{subject to} & x_i = \{0,\ 1\} \\ -x_1 & = \alpha \\ -x_7 & = \beta \\ x_1 - x_2 & = 0 \\ -x_2 + x_3 & = 0 \\ x_4 - x_8 & = 0 \\ x_5 - x_7 - x_8 & = 0 \\ x_6 & = 1 \\ -x_6 + x_8 & = 0 \\ x_2 - x_6 & = 0 \end{array}$$

where $\{\alpha, \beta\}$ can be $\{-1,0\}$, $\{-1,-1\}$, $\{0,0\}$, or $\{0,-1\}$.

After performing the tests again, only condition 1 can be solved under current structure. This new model rules out other three impossible conditions. As concluded, this modification improves the model's reliability.

V. CONCLUSIONS

In this paper, reachability analysis based model validation problem is discussed. Petri net is used to model biological systems. An investigation on the existence of nonnegative integer solutions of state equations introduces a new angle to rule out impossible biological states and further, check and improve the model structure. Two methods are proposed to do the verifications and two subsequent case studies exemplify the successful applications in biological systems.

REFERENCES

[1] H. Kitano, Systems Biology: A Brief Overview, *Science*, vol. 295, 2002.
[2] H. Kitano, Computational Systems Biology, *Nature*, vol. 420, 2002.
[3] H. D. Jong, Modeling and Simulation of Genetic Regulatory Systems: A Literature Review, *Journal of Computational Biology*, vol. 9, no. 1, 2002, pp 67-103.
[4] Y. Yang, K.S. Lee, C. Xiang and H. Lin, Biological mechanisms revealed by a mathematical model for p53-Mdm2 core regulation, *IET Systems Biology*, vol. 3, no. 4, 2009, pp 229-238.
[5] T. Murata, Petri Nets: Properties, Analysis and Applications, *Proceedings of the IEEE*, vol. 77, no. 4, 1989, pp 541-580.
[6] A. Ichikawa and K. Hiraishi, A class of Petri nets that a necessary and sufficient condition for reachability is obtainable, *Trans. Society of Instrument and Control Engieers (SICE)* (in Japanese), vol. 24, no. 6, 1988.
[7] G. A. Jones and J. M. Jones, *Elementary Number Theory*, Berlin: Springer-Verlag, pp. 7-11, 1998.
[8] Wikipedia, Smith normal form, http://en.wikipedia.org/wiki/Smith_normal_form
[9] M. Newman, *Integral Matrices*, Academic Press, 1972.
[10] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & sons, 1998.
[11] http://www.lindo.com/
[12] B. Alberts, D. Bray, J. Lewis, K. Roberts and J. Watson, *The Molecular Biology of the Cell*, 3rd Edition, Garland Publishing, New York; 1994.