Towards a Formal Fuzzy Framework for Modeling and Stationary Analysis of GRNs

Raed I. Hamed

Abstract: Many different approaches have been developed to model and stationary analysis of gene regulatory networks (GRNs). Recently, synthesis of GRNs which has desired functions is of interest to many researchers because it is a complementary approach to understanding GRNs, and it can be the first step to controlling living cells. In this paper, a fuzzy Petri net (FPN) approach to modeling and stationary analysis of fuzzy rule-based reasoning is proposed to handle uncertain and imprecise information. FPN structure used to represent knowledge and analyze the behavior of stationary states of gene regulatory networks, which integrates fuzzy, active and deductive rules with gene networks describing the flower morphogenesis of A. thaliana. This shows that the proposed method can be successfully applied in the study of biological systems.

Keywords: Fuzzy Petri net; stationary state; arabidopsis thaliana; Modeling and analysing.

I. INTRODUCTION

Recently there have been increasing research interests in synthesizing gene regulatory networks and several studies have been done [4, 10, 12]. Those studies are motivated by two ways. One is that the stationary analysis of gene regulatory networks having desired properties is a complementary approach to investigating mechanism and functions of gene regulatory networks. The other is that the modeling and stationary analysis of gene regulatory networks can be the first step to the control problem of living cells.

A major challenge of modeling biological systems is that conventional methods based on physical and chemical principles require data that is difficult to accurately and consistently obtain using either conventional biochemical or high throughput technologies, which typically yield noisy, semi-quantitative data (often in terms of a ratio rather than a physical quantity) [1, 4]. The problem of modeling is concerned with searching for optimal (or near-optimal) models subject to a number of constraints.

Petri net theory [2] and fuzzy logic exhibit a graphical and mathematical formalism to model, and simulate the biological systems. Fuzzy Petri net (PN) is a successful tool for describing and studying information systems. Incorporating the fuzzy logic with Petri nets has been widely used to deal with fuzzy knowledge representation and reasoning [5, 8].

One of central goals of contemporary molecular biology is to understand the cellular processes described by time series of thousands of gene expression measurements. The interactions between genes can be represented in a graphical form as a gene regulatory network. Some previous work on applications of Petri nets to regulatory networks can be found in [3]. The crucial task in modeling gene regulation is the analysis of stationary states of the network. Multistationarity (i.e. the property of systems whose structure induces two or more distinct steady states) can account for epigenetic differences also those involved in cell differentiation [9]. Another interesting approach to the stationary analysis based on structural properties of the interaction system can be found in [10]. With this processing paradigm, FPN models are inherently able to represent and process uncertainty and imprecision information, things which are quite evident in all aspects of the real world problems. Logical and fuzzy Petri nets seem to be a good choice for knowledge representation and reasoning in GRNs, where the numerical values for the parameters characterizing the interactions and the concentrations are most often imprecise and fuzzy information. The fuzzy Petri net approach was applied in [5, 6, 7], where authors exploit the well known notion of FPNs. In
contrast to previous methods we propose a fully automatic effective procedure for analyzing the stationary states of the network. Our definition of the notion of stationarity reflects the gene expression of the system that underlies the gene regulation process.

The organization of this paper is as follows: In Section 2, a knowledge representation scheme based on fuzzy Petri nets is described. In Section 3, Petri nets as fuzzy modeling are presented. In Section 4, we explain the details of the stationary state analysis are investigated in this Paper. Finally, we presented the conclusions of my model in Section 5.

II. A KNOWLEDGE REPRESENTATION SCHEME BASED ON FUZZY PETRI NETS

Production rules (PRs) are suitable to express expert knowledge. In most cases, collecting data in a precise way is difficult, FPRs are thus adopted, which have the ability of process uncertain or incomplete knowledge. For these reasons, inference rules are obtained in the form of FPRs, enhancing reasoning capacity. FPNs are built on the basis of FPRs. The knowledge representation scheme based on the FPNs theory is defined as 11-tuple [8]: KFPN = \((P, T, I, O, M, \Omega, \mu, f, c, \alpha, \beta)\), where \(P, T, I, O, M, \Omega, \mu, f\) and \(c\) are components of a generalized FPN as follows:

- \(P = \{p_1, p_2, ..., p_n\}\) is a finite set of places,
- \(T = \{t_1, t_2, ..., t_m\}\) is a finite set of transitions,
- \(I : T \rightarrow P^\infty\) is an input function, a mapping from transitions to bags of places,
- \(O : T \rightarrow P^\infty\) is an output function, a mapping from transitions to bags of places,
- \(M = \{m_1, m_2, ..., m_r\}\), \(1 \leq r < \infty\), is a set of tokens,
- \(\Omega : P \rightarrow P(M)\) is a mapping, from \(P\) to \(P(M)\), called a distribution of tokens, where \(P(M)\) denotes the power set of \(M\). By \(\omega0\) we denote the initial distribution of tokens in places of the FPN.
- \(\mu : P \rightarrow N\) is a marking, a mapping from places to non-negative integers \(N\). A mapping \(\mu\) can be represented as an \(n\)-component vector \(\mu = (\mu_1, \mu_2, ..., \mu_n)\), where \(n\) is a cardinality of the set \(P\). Obviously, \(\mu(p_i) = \mu_i\) and \(\mu(p_i)\) denotes the number of tokens in the place \(pi\). An initial marking is denoted by \(\mu_0\).
- \(f : T \rightarrow [0, 1]\) is an association function, a mapping from transitions to real values between zero and one.
- \(c : M \rightarrow [0, 1]\) is an association function, a mapping from tokens to real values between zero and one.

The complete information about the token \(mi\) is given by a pair \((p_i, c(m_i))\), where the first component specifies the location of the token, and the second one its value.

The bijective function \(\alpha : P \rightarrow D\) maps a set of places \(P\) into a set of concepts \(D\). The set of concepts \(D\) consists of the formal objects used for representing objects and facts from the agent’s world. The elements from \(D = D_1 \cup D_2 \cup D_3\) are as follows: elements that denote classes or categories of objects and represent higher levels of abstraction \((D_1)\), elements corresponding to individual objects as instances of the classes \((D_2)\) and those elements representing intrinsic properties of the concepts or values of these properties \((D_3)\).

The subjective function \(\beta : T \rightarrow \Sigma\) associates a description of the relationship among facts and objects to every transition \(t_i \in T; i = 1, 2, ..., m\). The set \(\Sigma = \Sigma_1 \cup \Sigma_2 \cup \Sigma_3\) consists of elements corresponding to the relationships between concepts used for partial ordering of the set of concepts \((\Sigma_1\)). The elements used to specify types of properties to which values from subset \(D_3\) are assigned \((\Sigma_2)\), and the elements corresponding to relationships between the concepts, but not used for hierarchical structuring \((\Sigma_3)\). For example, elements from \(\Sigma_3\) may be used for specifying the spatial relations among the objects. The functions \(\alpha\) and \(\beta\) give semantic interpretations to the scheme.

The inverse function \(\alpha^{-1} : D \rightarrow P\), and the generalized inverse function \(\beta^{-1} : \Sigma \rightarrow \tau\); \(\tau \subseteq T\) are defined in the KFPN scheme. The knowledge scheme KRFP can be graphically represented in a similar way to the Petri nets: circles represent places, while bars are used for the transitions. The relationships from places to transitions and from transitions to places are represented by directed arcs. Each arc is directed from an element of one set \(P\) or \(T\) to an element of another set \((T\) or \(P\). The relationships between elements from \(P\) and \(T\) are specified by the input and output functions \(I\) and \(O\), respectively. The tokens in the KFPN are represented by labeled dots: \(\bullet c(m_i)\). Denoting the places by elements of \(D\), the transitions by elements of \(\Sigma\) and the values of the association function by \(f\), the graphical representation of a knowledge base designed by the KFPN is obtained. Tokens give dynamical properties to the KFPN, and they are used to define its execution, i.e., by firing an enabled
transition \( t_j \) tokens are removed from its input places (elements in \( I(t_j) \)). Simultaneously, new tokens are created and distributed to its output places (elements of \( O(t_j) \)). In the KFPN, a transition is enabled if each of its input places has at least as many tokens in it as arcs from the place to the transition and if the values of the tokens \( c(m_i), i = 1, 2, \ldots \) exceed a threshold value \( \lambda \in [0, 1] \). The number of tokens at the input and output places of the fired transition is changed in accordance with the basic definition for the original PN. The new token value in the output place is obtained as \( c(m_i) \cdot f(t) \), where \( c(m_i) \) is the value of a token at the input place \( p_i \in I(t) \) and \( f(t) \) is a degree of the truth of the relation assigned to the transition \( t_j \in T \).

Fig. 1 illustrates the firing of the enabled transition of the KFPN. The inference procedures - inheritance, intersection search and recognition – defined for the KFPN, use its dynamical properties.

![Firing an Enabled Transition](image)

**Figure 1:** Firing an Enabled Transition. *Left:* Before Firing \( c(mi) > \lambda \). *Right:* After firing \( c(mp) = c(mi) \cdot f(t) \)

### III. PETRI NETS AS FUZZY MODELING

The improved inference models provide a new mechanism and approach for forward reasoning. FPNs are extensively used for reasoning in practical application. We can construct a fuzzy system for the gene regulatory mechanism to analysis the stationary states of *A. thaliana* flower cells [11, 12, 13], as shown in Fig. 2, we use \( g(t) \) to denote the desired amount protein of the gene regulatory network in sampling time \( t \).

![Gene Expression of Fuzzy Analysis System](image)

**Figure 2:** Gene Expression of Fuzzy Analysis System

The goal is to measure the gene products [i.e. \( \forall t, g(t) = 0 \)] when it initially starts with some nonzero angle off the vertical [i.e. \( \Theta(0) \neq 0 \)]. The inputs of fuzzy controller are \( x(t) \) and \( y(t) \), where \( x(t) \) is the error between \( g(t) \) and \( \Theta(t) \), and \( y(t) \) is change-in-error. The output of fuzzy controller is \( \Delta u(t) \), which is the increment of the control input \( u(t) \) to the gene expression.

(A) **Modeling Fuzzy Analysis System by Petri Net**

Fuzzification transforms a crisp variable \( x \) to a fuzzy set denoted by \( X \) defined on its universe of discourse. We have:

\[
F(x) = X
\]

(1)

where \( F \) is fuzzification operator. Since most practical work in fuzzy control system uses singleton fuzzification [14], we will also use it in this paper. Its membership function is defined by

\[
\mu_X(k) = \begin{cases} 
1 & k = x \\
0 & \text{otherwise} 
\end{cases}
\]

(2)

Suppose that the fuzzy set \( X \) is divided into \( 2m+1 \) fuzzy quantification degrees and expressed by linguistic value low, medium, high, respectively. Then, the fuzzification can be expressed by the following Petri net as shown in Fig. 3 (a). Since I use singleton fuzzification, one and only one transition of \( (t_{x_1}, t_{x_2}, t_{x_3}) \) is fired at one time. Suppose that \( t_{x_3} \) is fired, we have the result shown in Fig. 3 (b).

![Fuzzification Operation](image)

**Figure 3:** Fuzzification Operation (a) Before Firing of Any Transition, (b) After Firing of \( t_{x_3} \)

From Fig. 2, suppose that \( x(t) \) is fuzzified into \( M = 2m+1 \) fuzzy quantification degrees and \( y(t) \) into \( N = 2n+1 \) fuzzy quantification degrees. Then, there are \( M \times N \) rules for all possible cases in fuzzy rule-base. Suppose that \( y(t) \) is fuzzified into \( 2r+1 \) fuzzy quantification degrees. Then the fuzzy variables \( y_{ij} \) \( \in \{ \text{low}, \text{medium}, \text{high} \} \), where \( i = 1, 2, 3 \). They can be modeled by following Petri net Fig. 4:

In Fig. 4, one and only one transition is enabled in the situation of singleton fuzzification, that is to say, one and only one output place will get a token after firing. So far, we have used fuzzy logic to quantify the
rules in the rule-base, fuzzification to produce fuzzy sets characterizing the inputs, and the inference mechanism to fuzzy sets representing the conclusions that it reaches after considering the current inputs and the information in the rule-base. Next, we need to convert this fuzzy set quantification of the conclusions to a numeric value that can be input to the plant. All defuzzification methods are to choose a single output \( u(t) \) based on either the implied fuzzy sets or the overall implied fuzzy sets depending on the type of inference method chosen. They can be expressed in the same Petri net model Fig. 5. we adopt the “center of gravity” method as the defuzzification of the output predict change in target gene expression level. After get the crisp value of the output, we map it into its fuzzy membership and get the linguistic value whose membership degree has the highest level for change in target gene expression level.

This algorithm gets fuzzy Petri Net as an input and creates set of linguistic descriptions corresponding to each output place of fuzzy Petri Net:

**Algorithm 1. The Linguistic Descriptions of FPN Model**

**Input**: Fuzzy Petri Net: \( FPN \)

**Output**: set of linguistic descriptions: If Then

If Then = \( \emptyset \);

for each output place \( p_j \) of \( FPN \) do // create linguistic description

// create set of input variables \( (p_i) \) on whose \( p_j \) depends

inputs = \( \emptyset \);

for each input transition \( t_i \) of \( p_j \) do // add all inputs of transition \( t_i \) to inputs set

inputs = inputs \( \cup \) \( t_i \).inputs;

end

// construct the set of rules

\[ R = \emptyset \];

for each input transition \( t_i \) of \( p_j \) do // construct rule corresponding to transition \( t_i \)

\[ rule = \emptyset \];

for each element \( in \) from \( inputs \) do

if \( rule \neq 0 \) then rule = rule + AND;

if \( rule \in t_i\).inputs then

\[ rule = rule + rule.name \text{ is edge}(rule, t_i).value; \]

else

\[ rule = rule + rule.name \text{ is undefined}; \]

end

end

\[ rule = rule + THEN p_j.name \text{ is edge}(t_i, p_j).value; \]

\[ R = R \cup rule; \]

// add rule to rule base

end

If Then = If Then \( U \) \( R \) // add rule base to set of linguistic descriptions

end

---

**B. Petri Net Model of Gene Expression**

Fig. 6 illustrates the activate and repress processes, and then how we can map the relationship from biological
Towards a Formal Fuzzy Framework for Modeling and Stationary Analysis of GRNs

concept to knowledge fuzzy Petri net. Gene $a$ activates or represses gene $b$, as shown by the arrow in the upper part of the left- and right-hand side diagram, respectively. The bottom parts show how this is expressed in terms of fuzzy Petri nets.

![Gene Relationships and their Fuzzy Petri Nets Models](image)

Now, we can build fuzzy sets of $x(t)$, $y(t)$ and $\Delta u(t)$, and rule-base that inference mechanism works on. **Step 1**: we build membership function of the inputs $x(t)$ and $y(t)$ of the fuzzy controller in their universe of discourse as following forms.

$$
X = \begin{cases} 
  \text{Low} & x(t) < 0.5 \\
  \text{Medium} & x(t) = 0.5 \\
  \text{High} & x(t) > 0.5 
\end{cases}
$$

(4)

$$
Y = \begin{cases} 
  \text{Low} & y(t) < 0.5 \\
  \text{Medium} & y(t) = 0.5 \\
  \text{High} & y(t) > 0.5 
\end{cases}
$$

(5)

**Step 2**: we build membership function of the output $\Delta u(t)$ of the fuzzy controller in its universes of discourse. The fuzzy membership function of the output, i.e. changes in gene expression level is defined in Fig. 7. It is represented with five levels or five sets with respect to fuzzy theory, namely Very low, Low, Medium, High, and Very high.

![The Fuzzy Membership Function of Output $\Delta u(t)$](image)

**Step 3**: In the reasoning with fuzzy reasoning rules we build a rule-base for the gene expression, as shown in Fig. 8. In my algorithm, input genes are drivers. The heuristic rules are constructed accordingly based on a given data regulatory logic. For example, if the $X$ and the $Y$ genes are qualitatively classified as low, then the output expression level is considered medium. In another case, if the first is high and second is low, then the output expression level is high. Similar heuristics are applied to construct the decision matrix shown in Fig. 8. The input and output links represent antecedent and consequent parts of fuzzy rules, respectively. For instance, let us consider a simple structure which has only two input nodes, $x(t)$ and $y(t)$; nine rule nodes, $R_1, R_2, \ldots, R_9$; and one output node $u(t)$. Let the input and output links be represented by three linguistic terms: low is fuzzy set which represents “low expression level”, medium is fuzzy set which represents “medium expression level” and high is fuzzy set which represents “high expression level”. The reasoning engine performs decision-making based on the fuzzy logic reasoning rules with first order predicate logic. Each rule can be defined as an If-Then clause, which determines the linguistic value of output according to the linguistic values of input. Those fuzzy reasoning rules are shown as follow:

![Decision Matrix Describing a form of Fuzzy Rule Base](image)
The process is shown in Fig. 7 there is overlapping among adjacent fuzzy sets, so that more than one rule may be active for each expression value. For example in our model these rules can be used to predict the value of \( P_8 \):

\[
R1: \text{If } x(t) (d_1) \text{ is "Low" and } y(t) (d_4) \text{ is "Low" then } u(t) (d_8) \text{ is "Very low" (CF = } \mu = 0.89) \]

\[
R5: \text{If } x(t) (d_2) \text{ is "Medium" and } y(t) (d_5) \text{ is "Medium" then } u(t) (d_8) \text{ is "Medium" (CF = } \mu = 0.93) \]

\[
R9: \text{If } x(t) (d_3) \text{ is "High" and } y(t) (d_6) \text{ is "High" then } u(t) (d_8) \text{ is "Very high" (CF = } \mu = 0.95) \]

Finally, fuzzy Petri net model of the gene expression fuzzy control system shown in Fig. 2 can be expressed by the Fig. 9 according to the modeling fuzzy Petri net methods. Determine the marking value of the token after being passed to the output place (i.e. firing fuzzy production rules can be considered as firing transitions) according to the indicated operations required by the transition. For example, the following operations are executed by the AND and OR transitions.

**AND transition:**

\[
M'_{\Omega, \text{output\_place}} = \min_{\alpha, \text{input\_place}} M'_{\Omega, \alpha}, \quad (7)
\]

**OR transition:**

\[
M'_{\Omega, \text{output\_place}} = \max_{\alpha, \text{input\_place}} M'_{\Omega, \alpha}, \quad (8)
\]

where \( \Omega \) is the current set of the places which the token has traveled along, and the token value in a place \( p_i \), \( i \in P \), is denoted by \( \alpha(p_i) \in [0, 1] \).

Applying the maximum operation to all the resulting implications performs the aggregation. The linguistic terms of input and output nodes and the centers of the membership functions of linguistic terms (Triangular membership functions employed here) should be correctly determined in order for the fuzzy inference system to produce corresponding outputs according to inputs in training data. However, the following algorithm uses to construct fuzzy Petri net model.

**Algorithm 2. Constructing Fuzzy Petri Net Model**

Begin

Describe the knowledge base by a set of rules for practical problem.

While there are still some rules Do

Create a set \( T \) of transitions.

Set a certainty factor value (CF = \( m_j \)) for each transition \( T \).

Examine each antecedent-consequent proposition parts

For each one of them Do

Create a set of \( P \) of FPN places

Set a degrees of truth of Antecedent propositions, \( \alpha(p_i) = y_i \), and \( y_i \) is the degree of truth of proposition \( d_i \) given by user.

Create a set of \( I: (P \times T') \times [0,1] \) of input matrix define arcs from Antecedents to \( T \).

Create a set of \( O: (P \times T') \times [0,1] \) of output matrix define arcs from \( T \) to consequents

End For

End While

Construct fuzzy inference groups

End

![Figure 9: Fuzzy Petri Net Model of Gene Expression Fuzzy Analysis System](image-url)
IV. STATIONARY STATE ANALYSIS

The fuzzy analysis system shown in Fig. 2 is in an gene expression state if there exists nonnegative integer $T$, $x(t)$ is always fuzzified into $X = Z$ and $y(t)$ is always fuzzified into $Y = Z$, when $t > T$. The necessary and sufficient conditions under which the fuzzy analysis system is stable are: if there exists a nonnegative integer $T$, firing sequence is only $T = T_{s}, T_{x_{i}}, T_{x_{j}}, T_{u_{i}}$. The gene regulatory mechanism, that lay behind the flower morphogenesis of Arabidopsis thaliana are broadly studied by biologists. The mechanisms that control the differentiation of the cells are well understood. Extensive research allowed for construction of a gene regulatory network that depicts the mechanism governing this process. For this reason the model of A. thaliana is commonly used to test new mathematical formalisms for identifying stationary states see [11, 12, 13].

During the flower morphogenesis the cells are subject to a differentiation. This biochemical process ends when a chemical equilibrium is reached. The type of the equilibrium determines the type of the cell. The A. thaliana flower is build of four concentric whorls: sepals, petals, stamens and carpels. Research has shown that cells building up different parts of the flower have different types of chemical equilibrium. The ABC model is broadly used to describe these different equilibria. According to the model there are three types of activities (A, B and C) and their combination determines the type of the cell. Each activity has genes associated with it. The activity is observable if in the equilibrium these genes are active. Fig. 10 illustrates the connection between activities, genes and flower parts. Fig. 10 shows the graphical representation of the gene regulatory network underlying flower morphogenesis of A. thaliana. The vertices represent genes and arrows represent the regulatory relations. In the corresponding Petri net genes are represented by places and regulatory relations by transitions. After applying the fuzzy rules to the network from Fig. 10 we obtain a fuzzy Petri net output in table 1.

This section presents an analysis of the stationary states of A. thaliana flower cells using Petri nets, which have a uniform bound of 1 on all places. This models genes being either active (1) or inactive (0).

First we discuss the genes involved in the morphogenesis and later we show two different approaches to the analysis of the gene regulation network. Following [11, 12] we assume that the topology of the network is given. The first approach is to translate the network to the Petri net and then find all the stationary states. In the second approach we restrict the set of genes under question, translate to a smaller Petri net and find all the stationary states in that net. The first approach requires no preprocessing, but heavy post-processing is needed. The second approach requires one to restrict the set of analyzed genes, but yields results that require virtually no post-processing. Moreover, the first approach can lead to loss of some significant information due to lack of information on gene activity times. This is not the case in the second approach were the set of genes is by definition restricted to non-temporary ones. In this refined approach we consider the network model “from the developmental biology point of view” and highlight only the subset of interactions occurring in given nuclei in the particular time frame. Such an analysis is contrasted with “the functional genomics point of view” when regulations responsible for the all developmental processes are considered at the same time. The most studied example of the developmental gene regulatory network that controls the specification of endoderm and mesoderm in the sea urchin embryo is presented in [13].

The authors of [11, 12] use the following classification of the genes involved in the morphogenesis, with respect to times of their activity:
1. Genes active in the cells that are not part of the flower. During the morphogenesis they are active only in the very early stage. The genes are EMF1 and TFL1.

2. As the flower morphogenesis begins the expression of genes LFY and AP1 can be observed. The latter is also active in sepal cells of the flower.

3. Next, the genes LUG and UFO come to play, but their activity is only a part of the process and is not observed in the differentiated cells.

4. As the activity of the above mentioned genes fades away, the new genes responsible for the flower type - AG, AP3 and PI - become active.

5. At the end of the process the gene SUP is briefly active.

The genes can be divided into two groups, the genes that are temporary (LFY, LUG, UFO and SUP) and genes that are an essential part of the final equilibria (EMF1, TFL1, AP1, AG, PI and AP3). This classification allows one to distinguish between genuine and false stationary states, as it is unlikely for temporary genes to be observed in genuine stationary states.

<table>
<thead>
<tr>
<th>AG</th>
<th>PI</th>
<th>UFO</th>
<th>AP3</th>
<th>SUP</th>
<th>TFL1</th>
<th>LUG</th>
<th>EMF1</th>
<th>AP1</th>
<th>LFY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

V. CONCLUSIONS

The integration of fuzzy set theory with the Petri net modeling is an important step towards extending the modeling power of Petri Nets. With fuzzy Petri nets model introduced in this paper, uncertainties and different concurrent operations of a fuzzy analysis system can be modeled with a range of parameters using fuzzy membership functions. This paper proposed a fuzzy Petri net formalism for identification of stationary states analysis. The necessary and sufficient conditions under which the fuzzy analysis system is stable were derived by applying the fuzzy Petri nets model. We also showed that this approach yields very promising results when applied to a real world example such as the regulatory network of A. thaliana flower morphogenesis. This proposed method can decrease computation time and provide clear views of the stationary states of the regulatory network that can easily understand the inference process. In fuzzy analysis system modeling using fuzzy parameter sets, fuzzy membership functions are used to describe possible stationary states analysis that might exist and their effects over the behavior of the system. To illustrate the proposed method the system has applied to the real data sets and a satisfactory result was confirmed.

Acknowledgment

Many thanks for my colleagues for their good advices to this paper.

References


