Original Article

Conditional Petri Nets to Analyze and Simulate Automated Guided Vehicle System

Priya[1*], Sunita Kumawat[2]

[1*,2] Department of mathematics, Amity School of Applied Sciences, Amity University Haryana, Gurugram, India. E-mail:priyayadav17894@gmailcom,ksunita86@gmail.com

***Corresponding Author**:- Sunita Kumawat

*Department of mathematics, Amity School of Applied Sciences, Amity University Haryana, Gurugram, India. E-mail: ksunita86@gmail.com

DOI: 10.47750/pnr.2023.14.S01.73

Abstract

To examine the characteristics of various complex discrete event and distributed systems, Petri net is a potential mathematical and graphical modeling tool. The fuzzy production propositions of automated guided vehicle systems are represented in this paper using a variant of the Petri net known as the fuzzy Petri net (FPN), in which propositions explains the relation between two propositions. Due to its relevance in practice, another Petri net variant known as Conditional Petri net (CPN) is taken into consideration for this purpose. CPNs can be used to qualitatively describe gene regulatory interactions, to represent the active and inactive stages of a system like switching circuits, etc. The obtained truth degree of the success node validates the belief strength of this property, which is a crucial property of BPNs based on initial marking of Petri nets.

Keywords: Petri net, Fuzzy Petri net, Automated Guided Vehicle, Conditional Petri net, Simulation, Analysis.

1. INTRODUCTION

Petri nets have been used to model and analyze a variety of knowledge-based systems, including computational distributed systems, discrete event systems, communication protocols, manufacturing systems, performance evaluation, etc. Petri nets have also been used to model and research various kinds of biological networks. This is because they can represent the information flow in any expert system and model any graph-based structure. A German mathematician named Carl Adam Petri first introduced the idea of Petri nets in 1962 in his doctoral dissertation titled "Communication with Automata" that was submitted to Darmstadt University in West Germany. After that, they are utilized as a mathematical and graphical tool for modeling and studying synchronization, conflicts, and concurrency in distributed and discrete event systems.

Traditional PNs, however, are unable to accurately depict systems found in the real world. This is because some ambiguous and inaccurate information may be presented in these systems due to the ongoing expansion of the database of real-world knowledge systems. In order to model industrial control systems, Lipp [13] introduced a special subclass of Petri nets called Fuzzy Petri nets (FPNs) to handle uncertain or ambiguous information. Then, in 1988, Looney [15] modified PNs to represent rule-based expert systems using fuzzy reasoning and logic. Later, in 1990, Chen proposed a more general FPN model to model knowledge representation and also described a fuzzy reasoning algorithm that performs knowledge reasoning automatically. The time concept was supplemented with fuzzy Petri nets in [17], and the authors looked at how the time factor affected the net's performance in terms of transition firing and marking of the input and output locations. FPNs have been successfully used in a number of fields, including wireless sensor systems, operational management systems, biological networks, and fault diagnosis systems [3, 14]. In this paper, we use fuzzy propositions from the Conditional Petri nets (CPNs) subclass of safe Petri nets, a rule-based system. Petri nets modeling of Automated Guided Vehicle has been explained in [26, 27, 28]. In this paper, minimum spanning approach has been used to prevent from the deadlock. Conditional Petri nets are a new and promising class of Petri nets that are useful for building control systems, switching circuits, genetic regulatory networks, and many other kinds of systems [7, 21]. For instance, a gene can be active or inactive in a genetic regulatory network, meaning that there are two possible states for gene expression: ON (denoted by the letter "1") and OFF (denoted by the letter "0"). As a result, we can assign a Conditional variable to each gene that indicates whether or not it is active [21]. CPNs can therefore be used to describe and examine how genes interact with one another. Similar to analogue circuits, which are based on Conditional logic gates, digital circuits, also referred to as switching circuits, use signals with two different states (ON/OFF, 1/0) [6]. A helpful mathematical tool for streamlining switching circuits is conditional algebra. Therefore, using CPNs, the performance of these circuits can be easily examined. "If a Petri net is Conditional, then initial marking for all the places is one," the authors of [7] demonstrate. In this case, we are examining the reversal of the statement "IF initial marking for all the places is one, THEN it is a Conditional Petri net," that is, "IF initial marking for all the places is one, THEN a Petri net will generate all the results." It is accomplished by acquiring a sprouting tree that exhibits a relationship between. These assertions, as well as the degree to which they correspond to goals, are also calculated

using the starting proposition's degree of truth demonstrates the degree of conviction in the aforementioned property. In order to achieve this, the modified fuzzy reasoning algorithm proposed in [2] has been used. The obtained sprouted tree is then used to support an argument, and by using the available techniques for proof.
The remaining paper is organized in the manner shown here: Sect. 2 discusses fundamental ideas of Conditional Petri nets, fuzzy, Petri nets and all types of Petri nets. The fuzzified propositional algorithm is described in Section 3. The three cases used in Section 4 to illustrate how the algorithm is implemented. The formulation and validation of the argument are covered in Section 5. The conclusions and scope are provided in Section 6.

2. OVERVIEW OF PETRI NETS

Petri nets are classified as directed, weighted, bipartite multi-graphs with two different types of nodes: locations and transitions. When modeling conditions, places are represented by circles, while transitions are modeled by bars or rectangular boxes [8, 9, 12]. The condition-event (place-transition) net is another name for it [11]. Directed arcs are used to link these nodes together to represent the relationship between them. The place is known as the input (output) place of that transition if there is a directed arc connecting it to the other transition. Each arc has a positive integer assigned to it that represents its weight. In general, the weight is omitted for the arcs with weight 1. Tokens, which are nonnegative integers, are used to identify some locations. In a location, they are represented by using black dots. The number of tokens present in a location indicates the number of available data resources. The Petri net's state is represented by the marking vector M, which also provides the number of tokens that are present in each location within that state. PNs are described mathematically as [16]:

Definition 1: A 5-tuple is a Petri net: PN= (P; T; F; W; M_0); where P ={p_1, p_2,..., p_n}and T ={t_1, t_2,..., t_m }are respectively the finite non-empty set of places and transitions, with $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$, $F \subseteq (P * T) \cup (T * P)$ and $M_0 \rightarrow \{0,1,2, \dots\}$,

According to the transition firing rule, a Petri net is simulated. If all of a transition's input locations have more tokens than the weight of the arcs connecting them, the transition is considered enabled. Those locations serve as the transition's input, the enable tokens are used when an enabled transition fires. The transition consumes and deposits into each of its output locations in accordance with weight of the arc. A transition that has been fired alters the system's state. and a change in the marking as a result. You can find comprehensive information about PNs in [18]. The transition's firing rule is shown in Figure 1.

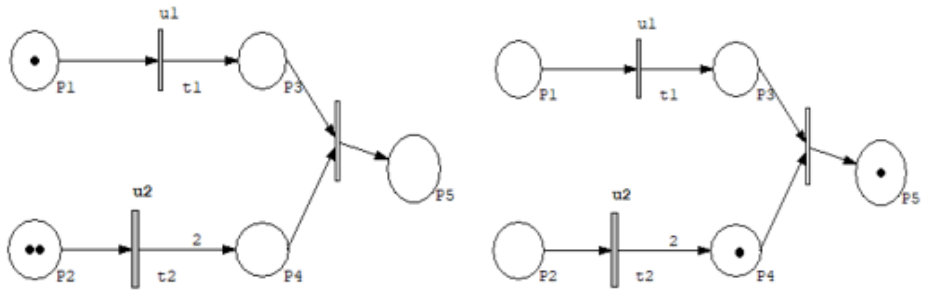


Fig. 1(a) Petri net before firing. **Fig. 1(b)** Petri net after firing.

Definition 2: The place p_j is referred to as being immediately reachable from the place p_i if p_i and p_j are, respectively, the input and output places of any transition between t_i; l=1, 2,..., m. The location p_k is referred to as reachable from the location p_i if it is another location that is immediately reachable from both p_j and p_i.
As seen in Fig. 1(a), for instance, place p_3 is immediately reachable from place p_1 and place p_4 is immediately reachable from place p_2, while place p_5 is also reachable from p_1 and p_2.

Definition 3: The place p_i's immediate reach-ability set, represented by IRS. The set of locations that can be reached right away from p_i is known as p_i. And the RS-indicated reach-ability set of p_i. P_i is the collection of locations that can be reached from p_i. For instance, in Fig.1(a)Petri net before firing, Places p_1 and p_2 are the transition's inputs, and p_3, $p_4$and p_5 are the transition's outputs. Here, t_1, t_2 and t_3 are enabled transition. (b) A Petri net following a shot. One token from p1, two tokens from p_2, and three tokens from p_4 are consumed and deposited on p_5 and the following firing of transition is followed t_1, t_2 and t_3. The initial marking is M_0={1,2,0,0,0} and the final marking after firing the transition is M_n={0,0,0,1,1}. The Petri net is said to be in a dead state at this point because no transition is enabled, meaning that no transition firing will occur.
IRS{p_1}={p_3}
IRS{p_2}={p_4}
IRS{p_3}={p_5}
IRS{p_4}={p_5}
RS{p_1}={p_5}

RS{p_2}={p_5}

Definition 4: If both of the locations p_i and p_j are input locations for the transition t_l, then they are referred to as adjacent locations with respect to the transition. For instance, the locations p_3 and p_4 in Fig. 1 are close by to the transition t_3.

Ordinary Petri nets, however, cannot accurately represent many real-world situations, and they are also unable to handle the ambiguous or uncertain data that most knowledge-based systems present. This is due to the fact that expert systems' databases are constantly growing, which causes these systems' complexity to rise.
Fuzzy Petri nets (FPNs), a Petri net extension for the knowledge representation of complex control processes and fuzzy rule-based expert systems, are created to address these drawbacks [13, 15].
A system with uncertainty is studied and modeled using a fuzzy Petri net. A truth value between 0 and 1 is linked to the token in a place and a certainty factor between 0 and 1 is linked to each transition in an FPN. Additionally, there can only be one token per location. Chen's definition of a fuzzy Petri net is [2]:
Definition 5: A fuzzy Petri net is an eight-tuple, which looks like this: $FPN = (P, T, Q, I, O, f, \alpha, \beta)$ where P, T and Q are non-empty finite sets of places, transitions, and propositions, respectively, $P \cap T \cap Q = \emptyset, |P| = |Q|$, Iand O are input and output functions, both from transition. An association function known as $\alpha: P \rightarrow [0,1]$connectseach transition to a certainty factor with a real value between 0 and 1. $\beta: P \rightarrow Q$is an association function also indicating the truth degree of e is [0,1]. Fig 2(a) shows the FPN model before firing the transition.
In an FPN, a transition t_i is enabled if $\forall\ p_j \epsilon\ I(t_i)$, meaning that for all input places p_j of the transition t_i, each place's truth degree is greater than or equal to the given threshold value k, i.e., $\alpha(p_j)$>k. The Fuzzy Production propositions are then fired, and each step in the reasoning process of an FPN updates the truth value of the output place.
A fuzzy relation between two propositions is defined by a fuzzy production proposition (FPP). Fuzzy 'IF-THEN' propositions are typically used to express FPPs, where the part of the proposition following 'IF' denotes the antecedent or precondition and the part of the proposition following 'THEN' denotes the consequent or post-condition [2, 15].
Let the two propositions of the FP_1 be If q_1 ("the machine is working") and q_2 ("You have required raw material on the working state"), THEN “the process will taken place and the product will be manufactured”. The two locations p_1 and p_2 will correspond to the two propositions q_1 and q_2, respectively. Let's say that the truth degree of the proposition q_1 is 0.90 and the threshold value is k= 0.52**.** Figure 2 depicts the modeling of FP_1 using FPN.

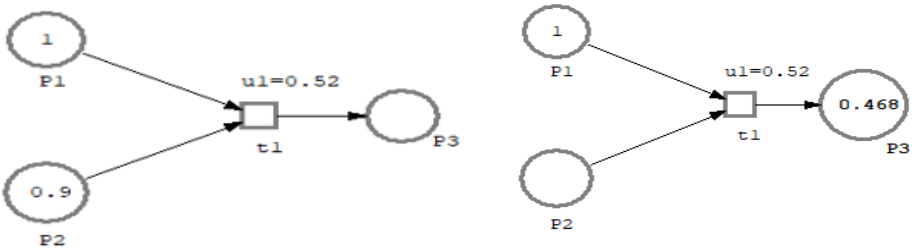


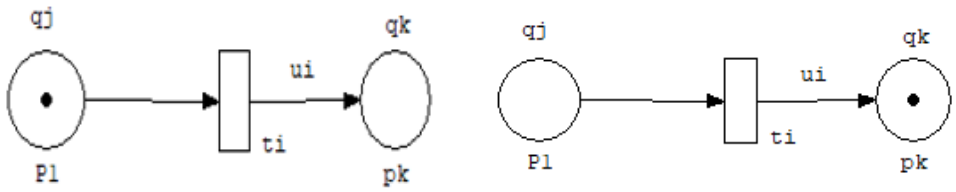
Figure 2(a): FPN before firing the transition,(b) after firing the transition.

In the FPN model mentioned above**,** P={p_1, p2 }

$$T = \{t_1\}$$
$$I(t_1) = \{p_1, p_2\}$$
$$O(t_1) = \{p_3\}$$
$$f(t_1) = \mu_1 = 0.52$$
$$\alpha(p_1) = 0.9$$
$$\alpha(p_2) = 1$$
$$\alpha(p_3) = 0$$
$$\beta(p_1) = q_1$$
$$\beta(p_2) = q_2$$
$$\beta(p_3) = q_3$$

Since $\gamma_1 > k$, thetruth degree of the proposition q_2 will be$\gamma_2 = \gamma_1 * \mu_1 = 0.468$shows in fig 2(b). It informs us that the product is manufactured with quantity 0.468. Above fig. depicts the firing of an FPN.
These are the most popular Fuzzy Production hypotheses: [2, 15]:

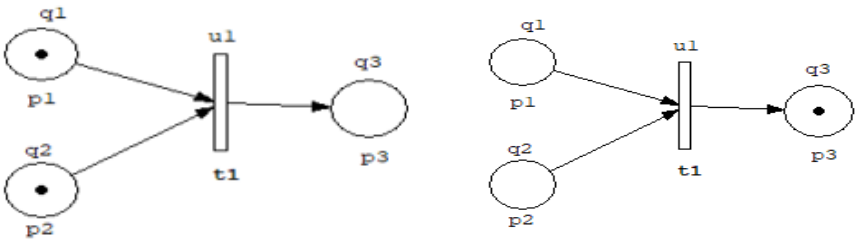
Type 1: If q_j then q_k



(a) Before firing the transition (b) After firing the transition

Figure 3: Type 1 of FPN

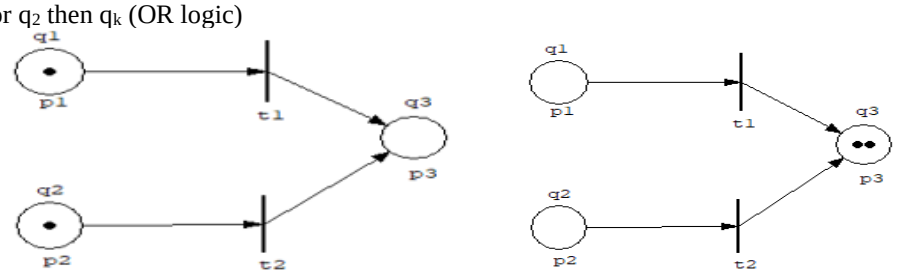
Type 2: If q_1 and q_2 then q_k (AND logic)



(a) Before firing the transition (b) After firing the transition

Figure 4: Type 2 of FPN (AND logic)

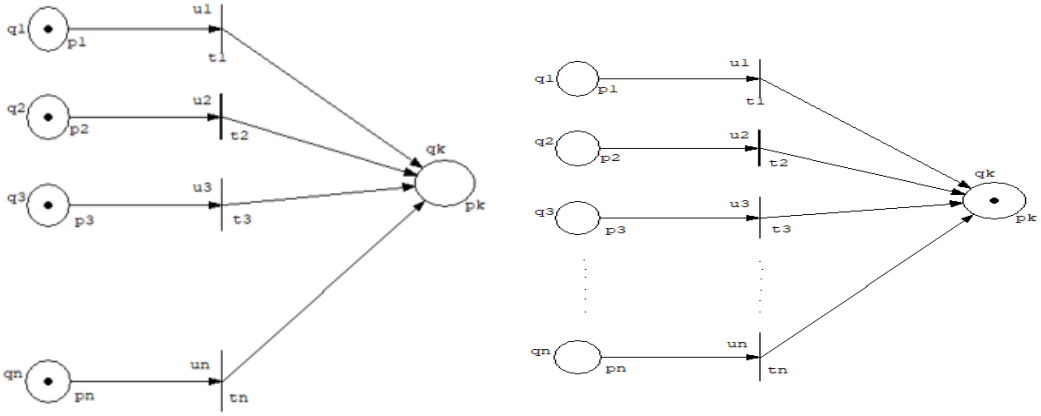
Type 3: If q_1 or q_2 then q_k (OR logic)



(a) Before firing the transition (b) After firing the transition

Figure 5: Type 3 of FPN (OR logic)

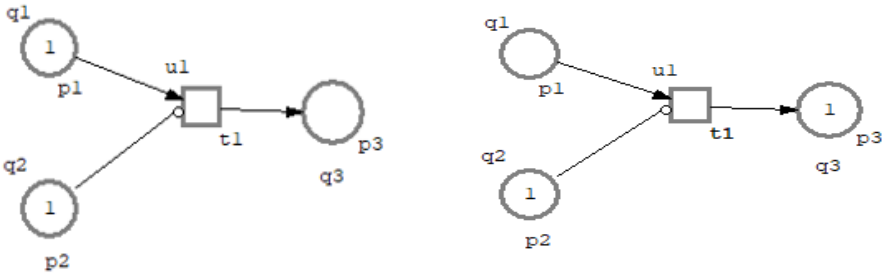
Type 4: If q_1 or q_2 or q_3 or …… q_n then q_k (R_{out} logic)



(a) Before firing the transition (b) After firing the transition

Figure 6: Type 4 of FPN (R_{out} logic)

Type 5: If q_1 and not q_2 then q_k (Inhibit logic)



(a) Before firing the transition (b) After firing the transition

Figure 7: Type 5 of FPN (Inhibit logic)

Conditional Petri nets, a special class of 1-safe Petri nets, were utilized in this study (CPNs). In their reach-ability tree, CPNs are 1-safe Petri nets that produce all n binary vectors as marking vectors [7-9, 22, 23]. See [7, 22] for more information on CPNs.

The rule-based CPN system's fuzzy production proposals have been used as input in this case. The truth degree of our goal proposition is computed using the fuzzified propositional algorithm, and three cases have been discussed by taking into account various fuzzy production proposition certainty factors [2].

3. PROPOSITIONAL FUZZIFIED ALGORITHM (FPA) FOR AGVS

This section has described a fuzzified propositional algorithm based on FPNs for AGV.

The FPA determines whether there is a precedent-subsequent relationship between the initial and goal propositions by using the truth degree of the initial proposition to automatically generate the truth degree of the goal proposition [2].

FPA creates a tree that lists every route from the starting point to the ending point. The 4-tuple $\{p_k, \gamma_k, IRS(p_k), \lambda\}$, where $p_k$denotes places, denotes a location, γ_k denotes the location's truth degree, $IRS(p_k)$ denotes immediate reach-ability set of p_k, and λ denotes the location's threshold. Let the initial and final locations, respectively, be p_i and p_f. As was previously stated, the places p_i and p_f will be related to the propositions q_i and q_f, respectively. Let's say that the user entered the proposition q_i, whose truth value is c_i, as the starting proposition. This means that the user wants to know what will be c_f, or the proposition q_f's truth degree.

A_{xy} denotes the set of adjacent locations of p_x, where p_y 2 IRS p_x, and $\mu_{xy} = a$denotes the certainty factor of the condition related to the propositions q_x and q_y. The following is a brief explanation of the fuzzified propositional algorithm.

Step 1: Starting with the first proposition, q_i, mark the tree's root node as$\{p_k, \gamma_k, IRS(p_k), \lambda\}and\beta(p_i) = q_s$. The non-terminal vertex is shown here.

Step 2: Pick a non-terminal vertex $p_a$in step two.

(a) If $IRS(p_a) = \phi\ then\ p_a$ is terminal vertex.

(b) If $IRS(p_a) \neq \phi\ and\ p_f \epsilon\ IRS(p_a)\ and\ \gamma_a \geq \lambda$, then make a new vertex p_f as $\{p_f, \gamma_f, IRS(p_f), \lambda\}$ where $\gamma_f = \gamma = \gamma_i * \mu_{af}$.

(c) If $p_f \notin IRS\ (p_a)\ and\ p_f \epsilon\ RS(p_a), then\ \forall p_n \epsilon\ IRS(p_a)$ lies between $p_a\ and\ p_f$ for reach-ability.

1) If $A_{ak} = \emptyset\ and\ \gamma_a > \lambda, and\ p_k$ does not held in any node in the way between root node p_i and the selected node p_a, then construct a new node p_k i.e., $\{p_k, \gamma_k, IRS(p_k), \lambda\}$ in the tree from the node $\{p_a, \gamma_a, IRS(p_a), \lambda\}$ to the node $\{p_k, \gamma_k, IRS(p_k), \lambda\}$ when $\gamma_k = \gamma_k * a$. Hence, $\{p_k, \gamma_k, IRS(p_k), \lambda\}$ is called the non-terminal node.

2) Else if

$A_{ak} = \{p_b, p_c, \ldots., p_z\}$i.e., then the decision maker will assign the truth degree of q_b, q_c,…., q_z for the node p_b, p_c,…., p_z respectively. Let us assume that t_b, t_c,….t_z be the truth degree entered by the decision maker(DM).

Let D=min (t_i, t_b, t_c,…..,t_z).

2.1) If $D \geq \lambda$, then make a new node p_k as $\{p_k, \gamma_k, IRS(p_k), \lambda\}$ in the tree from $\{p_a, \gamma_a, IRS(p_a), \lambda\}$ to $\{p_k, \gamma_k, IRS(p_k), \lambda\}$, where $\gamma_k = D * a$. Hence p_k will be called as non- terminal node.

2.2) else p_a will be assign as terminal node.

Step 3: If there is no non- terminal node left then go to step 4, else repeat step 2.

Step 4: If there is no process node, then there exist a relation between initial state and final state and the truth degree of the final state proposition will be solved as:

Let S be the set of success node i.e.,

$$S = \{(p_f, r_1, IRS(p_f), \lambda), (p_f, r_2, IRS(p_f), \lambda), \ldots\ldots, (p_f, r_n, IRS(p_f), \lambda)\}, 0 \leq r_i \leq 1, 1 \leq i \leq n.$$

Then set T=max (r_1,r_2,…,r_n). If the final proposition q_f has a truth degree of T, then the first and final propositions have a precedent-subsequent relationship of automated guided vehicle systems.

Else no relationship exists between these assumptions.

4. A CASE STUDY

In this section, we have made three different cases for AGV in Fuzzy Production propositions that correspond to CPNs. These claims were developed with the aid of [7, 8, 22]. The degree of each proposition's certainty varies in each scenario.

The following are the fuzzy production hypotheses for the rule-based Conditional Petri nets:

FP_1: IF $M_n(x) \leq 1 \forall x \epsilon P$ THEN Petri net is 1-bounded.

FP_2: IF the number of tokens in a place in PN never exceeds one THEN place is safe.

FP_3: IF all the places are safe or Petri net is 1-bounded, THEN Petri net is safe.

FP_4: IF a Petri net is safe and it generates all binary n-vectors as marking vectors in its reach-ability tree THEN Petri net is Conditional Petri net.

FP_5: IF a Petri net is Conditional Petri net, THEN $M_0(x) = 1\ \forall\ x\ \epsilon\ P, |P| \leq |T|$ and incidence matrix contain negative identity matrix of order n as a sub-matrix.

FP_6: IF a Petri net is safe and $M_0(x) = 1\ \forall\ x\ \epsilon\ P$ THEN it can be embedded as an included subnet of a CPN.

FP_7: IF $M_0(x) = 1\ \forall\ x\ \epsilon\ P$ THEN Petri net is a Conditional Petri net.

Let's refer to the aforementioned claims as:

$$q_1: M_n(x) \leq 1\ \forall\ x\ \epsilon\ P,$$
$$q_2: Petri\ net\ is\ 1 - bounded.$$
$$q_3: Number\ of\ tokens\ in\ a\ place\ in\ PN\ never\ exceeds\ one.$$
$$q_4: Place\ is\ safe.$$
$$q_5: Petri\ net\ is\ safe.$$
$$q_6:: Petri\ net\ generates\ all\ binary\ n\ vectors\ as\ marking\ vectors\ in\ its\ reachability\ tree.$$
$$q_7: Petri\ net\ is\ Conditional\ Petri\ net.$$

$$q_8: M_0(x) = 1 \; \forall \; x \; \epsilon \; P.$$
$$q_9: |P| \leq |T|.$$
$$q_{10}: Incidence\ matrix\ contains\ negative\ identity\ matrix\ of\ order\ n\ as\ a\ submatrix.$$
$$q_{11}: Petri\ net\ can\ be\ embedded\ as\ an\ induced\ subnet\ of\ a\ CPN.$$

Using the aforementioned notations to rewrite the FPPs, we have
FP_1: IF q_1 THEN q_2.
FP_2: IF q_3 THEN q_4.
FP_3: IF q_4 or q_5 THEN q_5.
FP_4: IF q_5 and q_6 THEN q_7.
FP_5: IF q_7 THEN q_8 and q_9 and q_{10}.
FP_6: IF q_5 and q_8 THEN q_{11}.
FP_7: IF q_8 THEN q_7.

The user is curious as to whether the assertions q_1 and q_7 have any sort of precedent-subsequent relationship. Because $\beta(p_1)=q_1$ and $\beta(p_7)=q_7$, our initial and goal propositions are q_1 and q_7, respectively, and the associated initial and goal placements are p_1 and p_7. Assume that in all three circumstances the threshold value, k, is equal to 0.30 and the truth degree, $\gamma$of the user-provided initial location, p_1, is equal to 0.90. Figure depicts the Fuzzy Petri net model of the FPPs of Conditional Petri nets. Tables 1 and 2, respectively, display the immediate reach ability set, reach ability set, and set of adjacent places. Let's talk about the cases:
Conditions for the following figures:
a) IF p_1 and p_2 THEN p_4.
b) IF p_2 and p_3 THEN p_5.

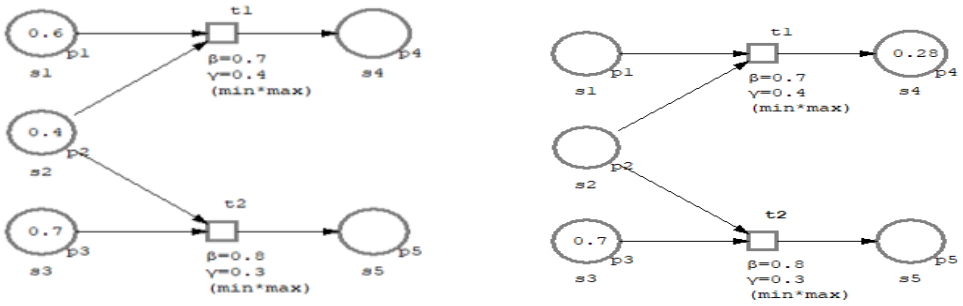
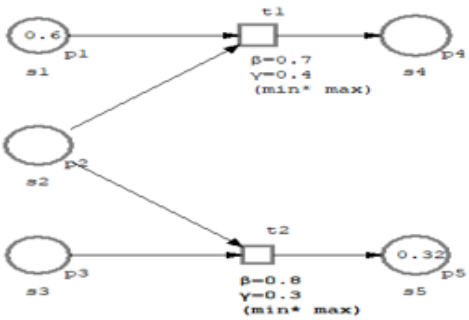


Figure 8: (a)Before firing the transition (b) After firing the transition t_1



(c) After firing the transition t_2

When t_1 fires according to FPN conditions:

Place (p_i)	IRS (p_i)	RS(p_i)	A_{ik}
P_1	{P_4}	{P_4}	{P_2}
P_2	{P_4}	{P_4}	{P_1}
P_3	Ø	Ø	Ø
P_4	Ø	Ø	Ø
P_5	Ø	Ø	Ø

When t_2 fires according to FPN conditions:

Place (p_i)	IRS (p_i)	RS(p_i)	A_{ik}
P_1	Ø	Ø	Ø
P_2	{P_5}	{P_5}	{P_3}
P_3	{P_5}	{P_5}	{P_2}
P_4	Ø	Ø	Ø
P_5	Ø	Ø	Ø

Reach-ability tree of above condition is shown in the following figure:

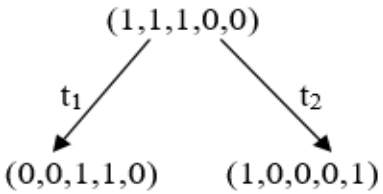


Figure 9: Reach-ability Tree

Cover-ability Tree for the above CPN is as follows:

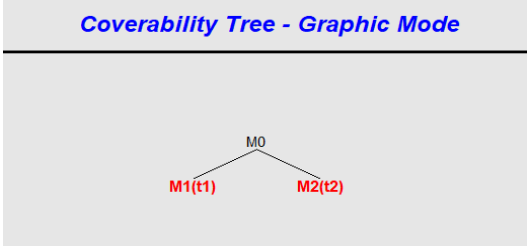


Figure 10: Cover-ability Tree

Case 1: In this case, the certainty factors are

$$\mu_1 = 0.90$$
$$\mu_2 = 0.85$$
$$\mu_3 = 0.80$$

The tree that was produced after applying the algorithm is displayed in Figure. Since we have a success node with a truth degree of $\gamma = 0.83$, it is clear that the first and last proposition are related in a precedent-subsequent manner. According to the truth degree value, there is a 0.83 percent chance that this relationship is true.

Case 2: Here,

$$\mu_1 = 0.83$$
$$\mu_2 = 0.78$$
$$\mu_3 = 0.70$$

we've lowered the values of the certainty factors. The newly adopted certainty factors are decreased. Following the implementation of the algorithm, Figure displays the resulting tree. Once more, we have a success node with truth degree γ = 0.63, indicating that the first and final propositions have a precedent-subsequent relationship. The lower value of truth degree demonstrates that as we reduce the FPPs' certainty factors, the likelihood that this relation is true also decreases; in this instance, it has become 0.63.

Case 3: In this instance, we further reduced the values of the certainty factors.

$$\mu_1 = 0.77$$
$$\mu_2 = 0.70$$
$$\mu_3 = 0.68$$

The newly adopted certainty factors are continuously decreased. Following the implementation of the algorithm, Figure displays the resulting tree. We have a success node again, but this time with truth degree $\gamma = 0.45$. The occurrence of the success node once more demonstrates that the first and last proposition have a precedent-subsequent relationship. The truth degree value has fallen even further, demonstrating that as we keep lowering the FPPs' certainty factors, the likelihood that this relation is true will also fall, and in this case, the likelihood has dropped to 0.45.

5. FORMULATION AND VALIDATION OF A PRODUCTION TREE IN AGVS

In this section, we'll develop an argument based on the tree we got in the previous section, then evaluate the validity of that argument.

After applying the fuzzified propositional algorithm in Section 4, we obtained tree for each case, but the path is

$$R_1: p_1 \rightarrow p_4$$
$$R_2: p_2 \rightarrow p_4$$
$$R_3: p_3 \rightarrow p_5$$
$$R_4: p_2 \rightarrow p_5$$

This line of reasoning leads to the following argument, which has four premises (R_1, R_2, R_3, R_4).

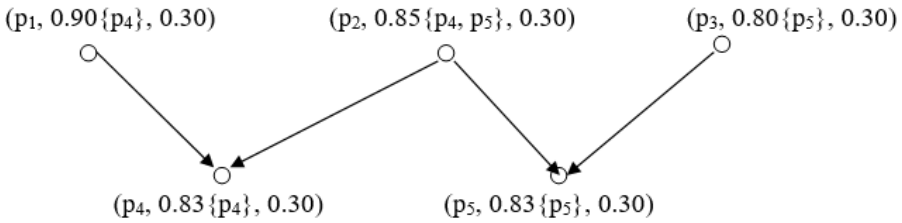


Figure 11: Tree for Case 1.

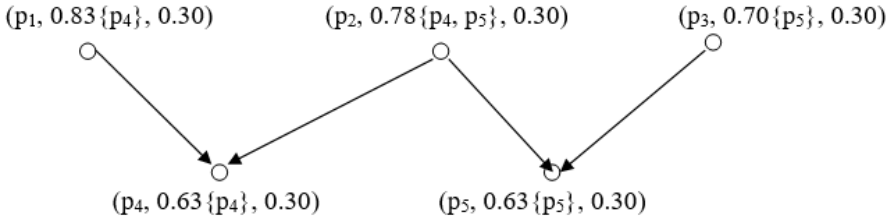


Figure 12: Tree for Case 2.

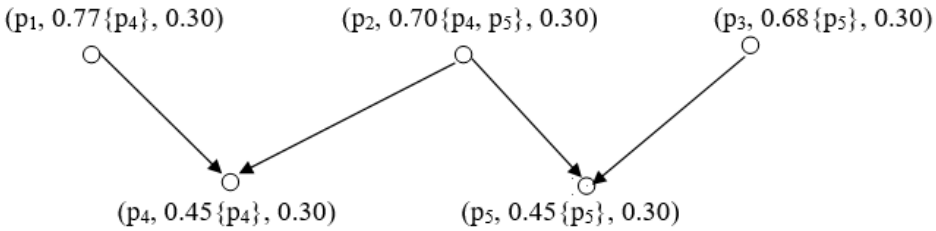


Figure 13: Tree for Case 3.

6. DISCUSSION AND GOALS

A precedent-subsequent relationship between the initial propositions, "$M_n(x) \leq 1 \; \forall \; x \; \epsilon \; P$," and the conclusion, "Petri net is a Conditional Petri net," has been demonstrated in this study. That is, "IF $M_n(x) \leq 1 \; \forall \; x \; \epsilon \; P$ THEN Petri net is a Conditional Petri net," and the argument validation proof also strongly supports this conclusion. However, the likelihood that this relation is true will vary depending on how certain the user's Fuzzy Production propositions are. Since the user chooses the certainty factors, higher values can be chosen, i.e. values close to 1, if we are aware that some propositions are true. According to the certainty factors, the possibility degree for the Fuzzy Production proposition "IF $M_n(x) \leq 1 \; \forall \; x \; \epsilon \; P$ THENPetri net is a Conditional Petri net" to be true in the first case is 0.83, while it decreases to 0.63 and 0.45 in the second and third cases, respectively. In particular, it will still hold if we assume that all places have initial markings of one, or $M_n(x) \leq 1 \; \forall \; x \; \epsilon \; P$**.** So, along with other conditions, we have demonstrated that the Petri net is a Boolean Petri net IF $M_n(x) \leq 1 \; \forall \; x \; \epsilon \; P$THEN.

Researchers are still working to solve the problem of computationally good characterization of Conditional Petri nets and its verification using Fuzzy Petri nets. The multi switches circuits used in AGVs in various applications may run more quickly and require less effort as a result of this characterization.

REFERENCES

1. Chaouiya, C.: Petri net modeling of biological networks. Brief. Bio inform. 8(4), 210–219, (2007)
2. Chen, S.M., Ke, J.S., Chang, J.F.: Knowledge representation using fuzzy Petri nets. IEEE Trans. Knowl. Data Eng. 2(3), 311–319 (1990)
3. Hamed, R.I., Ahson, S.I., Parveen, R.: A new approach for modelling gene regulatorynetworks using fuzzy Petri nets. J. Integr. Bio inform. 7(1), 113 (2010)
4. Heiner, M., Koch, I., Will, J.: Model validation of biological pathways using Petri nets demonstrated for apoptosis. Biosystems 75, 15–28 (2004)Hofestadt, R.: Advantages of Petri-net modeling and simulation for biological networks. Int. J. Biosci. Biochem. Bioinforma. 7(4), 221–229 (2017)
5. Jain, S., Naik, P.K., Bhooshan, S.V.: Petri nets – an application in digital circuits. In:
6. Proceedings of the ICWET 2010 International Conference & Workshop on Emerging Trends in Technology, p. 1005 (2010). https://doi.org/10.1145/1741906.1742168
7. Kansal, S., Acharya, M., Singh, G.P.: Boolean Petri nets. In: Pawlewski, P. (ed.) Petri nets Manufacturing and Computer Science, pp. 381–406. In-Tech Global Publisher (2012). ISBN 978-953-51-0700-2, Chapter 17
8. Kansal, S., Singh, G.P., Acharya, M.: On Petri nets generating all the binary n-vectors.
9. Scientiae Mathematicae Japonicae e-2010, 113–120 (2010). 71(2), 209–216
10. Kansal, S., Singh, G.P., Acharya, M.: 1-Safe Petri nets generating every binary n- vector exactly once. Scientiae Mathematicae Japonicae, e-2011, 127–137 (2011). 74(1) 29–36
11. Koch, I.: Petri nets in systems biology. Softw. Syst. Model. 14(2), 703–710 (2015)
12. Kumawat, S.: Weighted directed graph: a Petri net-based method of extraction of closed weighted directed euler trail. Int. J. Serv. Econ. Manag. 4(3), 252–264 (2012)
13. Kumawat, S., Purohit, G.N.: Total span of farm work flow using Petri net with resource sharing. Int. J. Bus. Process Integr. Manag. 8(3), 160–171 (2017)
14. Lipp, H.-P.: The application of a fuzzy Petri net for controlling complex industrial processes. In: Proceedings of IFAC Conference on Fuzzy Information Control, Marseille, pp. 459–465 (1983)

15. Liu, H.C., You, J.X., Li, Z.W., Tian, G.: Fuzzy Petri nets for knowledge representation and reasoning: a literature review. Eng. Appl. Artif. Intell. 60, 45–56 (2017)
16. Looney, C.G.: Fuzzy Petri nets for rule-based decision making. IEEE Trans. Syst. ManCybern. 18, 178–183 (1988)
17. Murata, T.: Petri nets: properties, analysis and applications. Proc. IEEE 77(4), 541–580 (1989)
18. Pedrycz, W., Camargo, H.: Fuzzy timed Petri nets. Fuzzy Sets Syst. 140(2), 301–330 (2003)
19. Peterson, J.L.: Petri Net Theory and the Modeling of Systems. Prentice-Hall, Englewood Cliffs (1981)
20. Petri, C.A.: Communication with automata. Ph.D. dissertation, Technical report. RADC-TR 65-377, Rome Air Development Center, Rome (1966)
21. Reddy, V.N., Mavrovouniotis, M.L., Liebman, M.N.: Petri net representations in metabolic pathways. In: Proceedings of the International Conference on Intelligent Systems for Molecular Biology (1993)
22. Remy, E., Mossé, B., Thieffry, D.: Boolean dynamics of compound regulatory circuits. In: Rogato, A., Zazzu, V., Guarracino, M. (eds.) Dynamics of Mathematical Models in Biology pp. 43–53. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45723-9_4
23. Singh, G.P.: Some advances in the theory of Petri nets. Ph.D. thesis, Faculty of Technology, University of Delhi, Delhi, India (2013)
24. Singh, G.P., Kansal, S., Acharya, M.: Embedding an arbitrary 1-safe Petri net in a Boolean Petri net. Int. J. Comput. Appl. 70(6), 7–9 (2013)
25. Zhou, K.Q., Zain, A.M.: Fuzzy Petri nets and industrial applications: a review. Artif. Intell. Rev. (2016). https://doi.org/10.1007/s10462-015-9451-9
26. Zhou, K.Q., Mo, L.P., Jin, J., Zain, A.M.: An equivalent generating algorithm to modelfuzzy Petri net for knowledge-based system. J. Intell. Manuf. 30(4), 1831–1842 (2019). https://doi.org/10.1007/s10845-017-1355-x
27. Priya, Sunita Kumawat: Petri net Modeling and Analysis of Automated Guided Vehicle System(2022). Journal of Rajasthan Academy of Physical Sciences, (25-36), ISSN: 0972-6306,http://raops.org.in.
28. S. Pal, Samrat Chatterjee, J. Chattopadhyay: Role of Toxin and nutrient for the occurance and termination of plankton bloom- Results drawn from field observations and a mathematical model. BioSystems 90(2007) 87-100.
29. Priya, Sunita Kumawat,"Petri Nets Modeling of Automated Guided Vehicle systems, Fault Detection and its Treatment", AIP Conference Procedings, April 2022. (Processing)