Fuzzy Reasoning Based On Petri Nets[1]

Changjun Jiang

Department of Computer Science and Engineering

Tongji University, Shanghai 200092

THE PEOPLE'S REPUBLIC OF CHINA

E-mail: cjjiang@online.sh.cn

Yingping Zheng

CIMS Center

Tongji University, Shanghai 20009

THE PEOPLE'S REPUBLIC OF CHINA

Abstract: In this paper, two new models - logical Petri net (LPN) and fuzzy logical Petri net (FLPN)- are defined. The fuzzy reasoning algorithm based on a sub-fuzzy logical Petri net is presented. This algorithm is backward reasoning according to goal propositions. It is simpler than the conventional algorithm of forward reasoning from initial propositions. The application of this model and the reasoning algorithm are discussed.

Key words: Logical Petri Net, Fuzzy Logical Petri Net, Semantics, Reasoning Algorithm

Changjun Jiang received his BE and ME from University of Shandong Science and Technology in 1986 and 1991, respectively. He received his Ph.D degree from Institute of Automation, Chinese Academy of Sciences. He is currently Professor at the Department of Computer Science and Engineering, Shanghai Tongji University. His areas of interest in research include Petri net, Concurrent system, Model checking and Fuzzy reasoning.

Yingping Zheng received his BE from Peking University in 1963 and graduated student from Institute of Automation, Chinese Academy of Sciences in 1967, respectively. He is currently Professor at the CIMS Center, Shanghai Tongji University. His areas of interest in research include Systems and Control Theory, e.g. Linear Systems, Optimal control, Differential game, DEDS and Hybrid systems, and their Applications.

1. Introduction

The problem of validation of knowledge bases is an important and difficult problem. Ordinary and high level Petri nets have been proposed as knowledge representation formalisms where structural and behavioral properties of the net can be used to prove properties of the system being modeled or to verify the knowledge base integrity [1-7]. [3] presented a validation method for Horn logic. The validation problems of Horn logic have been further researched in [4, 5, 6]. The G-net model was based on the knowledge table representation [7].

In a fuzzy environment, the issues of concern when talking about integrity checking are the definition of concepts such as inconsistency, redundancy and completeness, and the investigation of suitable similarity measures for comparison of fuzzy propositions [8-15]. One of the most successful applications of fuzzy logic has been in the area of processes control [16-18]. A generalized fuzzy Petri net model was introduced in [19]. A representational model for the knowledge base (KB) of fuzzy production systems with rule chaining based on the Petri net formalism was developed in [15]. The High-level fuzzy Petri net (HLFPN) model (introduced by the authors in [20,21]) as opposite to other approaches found in literature, derived from high-level Petri nets such as predicate/transition nets [22] and colored Petri nets [23]. In [24] a new reasoning algorithm for high-level fuzzy Petri nets was proposed. The Petri nets models and reasoning algorithms of neural nets were researched in [25,26]. In [27] the solving method of neural nets was given for problems of Petri nets.

This paper proposes that a new representational model for a kind of fuzzy production systems is established and the reasoning algorithms are presented based on Petri net. It is semantics which is mainly considered in this model. The algorithms procedures proposed in this paper are based on the results of applying the reflection on the input methodogy [15] to several cases of fuzzy inconsistencies, analysed in [21].

---

1 Supported by National Natural Science Foundation of P.R.China, National 973 Basic Foundation of P.R.China, Excellent Ph.D Paper Author Foundation of P.R.China, Dawn Plan Foundation of Shanghai and Excellent Young Scientist Foundation of Shandong Province.

The organization of the subsequent discussion is as follows. Section 2 introduces the basic notions of Petri net and the model of logical Petri net based on logical semantics (AND, OR, XOR). The model of fuzzy logical Petri net is presented in Section 3. In Section 4, the reasoning methods of reflection on the input proposed in [15] are summarized. In Section 5 a practical problem analysis is done by the given reasoning algorithm in Section 4.

2. Petri Nets and Logical Petri Nets

A triplet $N = (P, T; F)$ is called a net iff

(1) $P \cap T = \varnothing, P \cup T \neq \varnothing$;

(2) $F \subseteq (P \times T) \cup (T \times P)$; and

(3) $\mathrm{dom}(F) \cup \mathrm{con}(F) = P \cup T$.

For $\forall x \in P \cup T$, ${}^{\bullet}x = \{y \mid (y \in P \times T) \wedge ((y, x) \in F)\}$ and $x^{\bullet} = \{y \mid (y \in P \times T) \wedge ((x, y) \in F)\}$ are called the pre-set and the post-set of x, respectively.

Definition 1

5-tuple $LPN = (P, T; F, M_0, D, h)$ is called a Logical Petri net iff

(1) $N = (P, T; F)$ is a net;

(2) $P \cap D = T \cap D = \varnothing, |P| = |D|$;

(3) D is a finite set of propositions;

(4) $h: P \rightarrow D$ is an association function, representing a bijective mapping from places to propositions;

(5) $T = T^1 \cup T^{AND} \cup T^{OR} \cup T^{XOR}$, $\forall i, j \in \{1, AND, OR, XOR\}, i \neq j : T^i \cap T^j = \varnothing$;

(6) $\forall t \in T: |t^{\bullet}| = 1$; $\forall t \in T_1: |{}^{\bullet}t| = 1$; $\forall t \in T^{AND}: \underset{\forall p \in {}^{\bullet}t \wedge |{}^{\bullet}t| > 1}{AND} d(p)$; $\forall t \in T^{OR}:$

$\underset{\forall p \in {}^{\bullet}t \wedge |{}^{\bullet}t| > 1}{OR} d(p)$; $\forall t \in T^{XOR}: \underset{\forall p \in {}^{\bullet}t \wedge |{}^{\bullet}t| > 1}{XOR} d(p)$;

(7) $M_0 : P \rightarrow \{0, 1\}$ is called an initial marking of LPN;

(8) the following firing rules apply:

$t \in T, t$ belong to one of following four cases:

(8.1) $t \in T^1$, M is a marking of Σ, and t is said to be M-enabled (denoted as $M[t>$) if

$p \in {}^{\bullet}t \cap P: M(p) = 1 \wedge p \in t^{\bullet} \cap P: M(p) = 0$;

(8.2) $t \in T^{AND}$, M is a marking of Σ, and t is said to be M-enabled (denoted as $M[t>$) if

$\forall p \in {}^{\bullet}t \cap P: M(p) = 1 \wedge p \in t^{\bullet} \cap P: M(p) = 0$;

(8.3) $t \in T^{OR}$, M is a marking of Σ, and t is said to be M-enabled (denoted as $M[t>$) if

$\exists p \in {}^{\bullet}t \cap P: M(p) = 1 \wedge p \in t^{\bullet} \cap P: M(p) = 0$; and

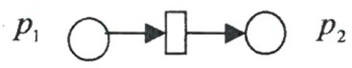
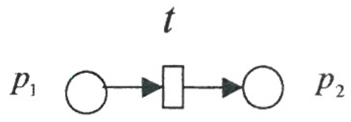
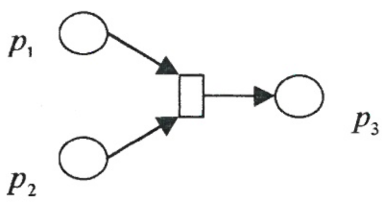
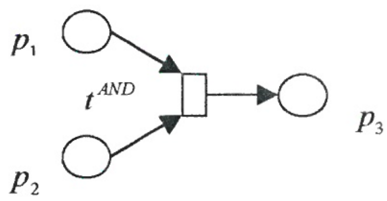
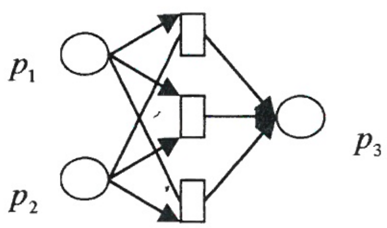
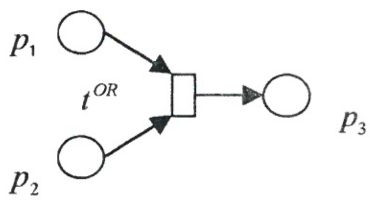
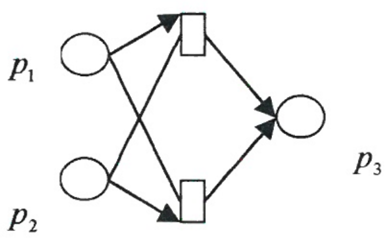
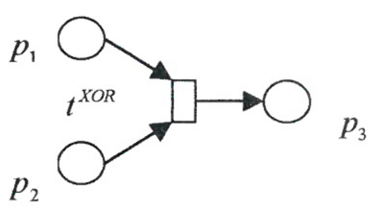
(8.4) $t \in T^{XOR}$, M is a marking of Σ, and t is said to be M-enabled (denoted as $M[t>$) if

there is only one $p \in {}^{\bullet}t \cap P: M(p) = 1 \wedge p \in t^{\bullet} \cap P: M(p) = 0$;

t can be fired from M if t is M-enabled (denoted as $M[t>$). Firing t from M results in a new marking M' (denoted as $M[t>M'$). For $\forall p \in P$, we have

$$M'(p) = \begin{cases} M(p)+1, & if\ p \in t^{\bullet} - {}^{\bullet}t; \\ M(p)-1, & if\ p \in {}^{\bullet}t - t^{\bullet}; \\ M(p), & otherwise. \end{cases}$$

Table 1. The Petri Nets Description of Logical Rules

No.	Rules	Petri Nets Semantics	Logical Petri Nets
1	IF d_1 THEN d_2		
2	IF d_1 AND d_2 THEN d_3		
3	IF d_1 OR d_2 THEN d_3		
4	IF d_1 XOR d_2 THEN d_3		

3. Fuzzy Logical Petri Nets

Definition 2

9-tuple $FLPN = (P, T; F, M_0, D, h, \alpha, \theta, \lambda)$ is called a Fuzzy Logical Petri net iff

(1) $LPN = (P, T; F, M_0, D, h)$ is a Logical Petri net;

(2) $\alpha : P \to [0, 1]$ is an association function, representing a mapping from place to real values between 0 and 1;

(3) $\theta, \lambda : T \to [0, 1]$ are association functions, representing a mapping from transition to real values between 0 and 1;

(4) the following firing rules apply:

(4.1) $t \in T^1$, M is a marking of Σ, and *t* is said to be *M*-enabled (denoted as $M[t>$) if $p \in {}^{\bullet}t \cap P: M(p)=1 \wedge p \in t^{\bullet} \cap P: M(p)=0$; and *t* can be fired from *M* if *t* is *M*-enabled (denoted as $M[t>$). Firing *t* from *M* results in a new marking *M'* (denoted as $M[t>M'$). For $\forall p \in P$, we have

$$M'(p)=\begin{cases} M(p)+1, & if\ p \in t^{\bullet}-{}^{\bullet}t; \\ M(p)-1, & if\ p \in {}^{\bullet}t-t^{\bullet}; \\ M(p), & otherwise, \end{cases}$$

$$\alpha(p)=\lambda_t \alpha(p') \quad if\ \alpha_1 \geq \theta_t \wedge p \in t^{\bullet} \wedge p' \in {}^{\bullet}t;$$

(4.2) $t \in T^{AND}$, M is a marking of Σ, and *t* is said to be *M*-enabled (denoted as $M[t>$) if $\forall p \in {}^{\bullet}t \cap P: M(p)=1 \wedge p \in t^{\bullet} \cap P: M(p)=0$; and *t* can be fired from *M* if *t* is *M*-enabled (denoted as $M[t>$). Firing *t* from *M* results in a new marking *M'* (denoted as $M[t>M'$). For $\forall p \in P$, we have

$$M'(p)=\begin{cases} M(p)+1, & if\ p \in t^{\bullet}-{}^{\bullet}t; \\ M(p)-1, & if\ p \in {}^{\bullet}t-t^{\bullet}; \\ M(p), & otherwise, \end{cases}$$

$$\alpha(p)=\lambda_t \min_{\forall p' \in {}^{\bullet}t} \alpha(p') \qquad if\ \min_{\forall p' \in {}^{\bullet}t}\{\alpha(p')\} \geq \theta_t \wedge p \in t^{\bullet};$$

(4.3) $t \in T^{OR}$, M is a marking of Σ, and *t* is said to be *M*-enabled (denoted as $M[t>$) if $\exists p \in {}^{\bullet}t \cap P: M(p)=1 \wedge p \in t^{\bullet} \cap P: M(p)=0$; and *t* can be fired from *M* if *t* is *M*-enabled (denoted as $M[t>$). Firing *t* from *M* results in a new marking *M'* (denoted as $M[t>M'$). For $\forall p \in P$, we have

$$M'(p)=\begin{cases} M(p)+1, & if\ p \in t^{\bullet}-{}^{\bullet}t; \\ M(p)-1, & if\ p \in {}^{\bullet}t-t^{\bullet}; \\ M(p), & otherwise, \end{cases}$$

$$\alpha(p)=\lambda_t \max_{\forall p' \in {}^{\bullet}t} \alpha(p') \qquad if\ \max_{\forall p' \in {}^{\bullet}t}\{\alpha(p')\} \geq \theta_t \wedge p \in t^{\bullet};$$ and

(4.4) $t \in T^{XOR}$, M is a marking of Σ, and *t* is said to be *M*-enabled (denoted as $M[t>$) if there is only one $p \in {}^{\bullet}t \cap P: M(p)=1 \wedge p \in t^{\bullet} \cap P: M(p)=0$; and t can be fired from M if t is M-enabled (denoted as $M[t>$). Firing t from M results in a new marking M' (denoted as $M[t>M'$). For $\forall p \in P$, we have

$$M'(p)=\begin{cases} M(p)+1, & if\ p \in t^{\bullet}-{}^{\bullet}t; \\ M(p)-1, & if\ p \in {}^{\bullet}t-t^{\bullet}; \\ M(p), & otherwise, \end{cases}$$

$$\alpha(p)=\lambda_t \alpha(p') \qquad if\ \alpha(p') \geq \theta_t \wedge there\ is\ only\ one\ p' \in {}^{\bullet}t \wedge p \in t^{\bullet}.$$

Definition 3

Let $FLPN=(P,T;F,M_0,D,h,\alpha,\theta,\lambda)$ be a Fuzzy Logical Petri net. Set $SFLPN=(P_s,T_s;F_s,M_{0s},D_s,h_s,\alpha_s,\theta_s,\lambda_s)$, such that

(1) $P_s \subseteq P$;

(2) $T_s \subseteq T$;

(3) $F_s = F \cap ((P_s \times T_s) \cup (T_s \times P_s))$;

(4) $M_{0s}: P_s \to \{0,1\}$, such that $M_{0s}(p)=M_s(p), \forall p \in P_s$;

(5) $D_s \subseteq D$ is a finite subset of sets of propositions;

(6) $h_s: P_s \to D_s$ is an association function, representing a bijective mapping from places to propositions, such that $h_s(p)=h(p), \forall p \in P_s$;

(7) $\alpha_s: P_s \to [0,1]$ is an association function, representing a mapping from places to real values between 0 and 1, such that $\alpha_s(p)=\alpha(p), \forall p \in P_s$;

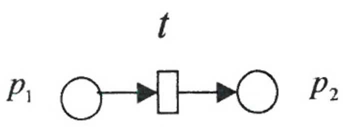
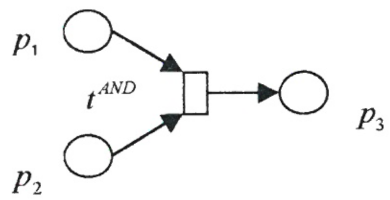
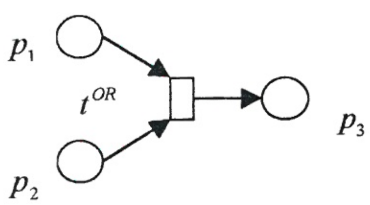
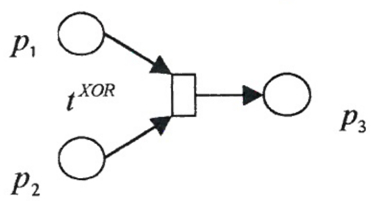
(8) $\theta_s, \lambda_s: T_s \to [0,1]$ are association functions, representing a mapping from transition to real values between 0 and 1, such that $\theta_s(t)=\theta(t), \lambda_s(t)=\lambda(t), \forall t \in T_s$;

(9) The firing rules are the same as in Definition 2.

Definition 4

Let $FLPN=(P,T;F,M_0,D,h,\alpha,\theta,\lambda)$ be a Fuzzy Logical Petri net. Set $P_0=\{p \mid M_0(p)>0 \wedge \forall p \in P\}$, then P_0 is called places set of initial true propositions. D_0 corresponding with P_0 is called a set of initial true propositions.

Table 2. The Fuzzy Computing Formulation of Fuzzy Logical Rules

No.	Rules	Logical Petri Nets	Fuzzy Computing
1	IF d_1 THEN d_2		$\alpha_2 = \lambda_t \alpha_1 \quad if\ \alpha_1 \ge \theta_t$
2	IF d_1 AND d_2 THEN d_3		$\alpha_3 = \lambda_{t_{OR}} \min_{\substack{\alpha_i \ge \theta_{t_{OR}} \\ i=1\wedge 2}} \{\alpha_1, \alpha_2\}$
3	IF d_1 OR d_2 THEN d_3		$\alpha_3 = \lambda_{t_{OR}} \max_{\substack{\alpha_i \ge \theta_{t_{OR}} \\ i=1\vee 2}} \{\alpha_1, \alpha_2\}$
4	IF d_1 XOR d_2 THEN d_3		$\alpha_3 = \begin{cases} \lambda_{t_{XOR}} \alpha_1 \ if\ \alpha_1 \ge \theta_{t_{XOR}} \wedge \alpha_2 = 0 \\ \lambda_{t_{XOR}} \alpha_2 \ if\ \alpha_2 \ge \theta_{t_{XOR}} \wedge \alpha_1 = 0 \end{cases}$

4. The Reasoning Algorithm of Fuzzy Logical Petri Nets

Algorithm *SSFLPN* (Spanning Sub-Fuzzy Logical Petri Net)

Input: $FLPN = (P, T; F, M_0, D, h, \alpha, \theta, \lambda)$, $P_g \subseteq P$.

Output: $SFLPN = (P_s, T_s; F_s, M_{0s}, D_s, h_s, \alpha_s, \theta_s, \lambda_s)$.

Step 1 Set $P_s \leftarrow P_g$; $T_s \leftarrow \varnothing$; $F_s \leftarrow \varnothing$; $P_{0s} \leftarrow \varnothing$;

Step 2 Let each p_g into stack S, for $\forall p_g \in P_g$;

Step 3 If $S \ne \varnothing$ then let top element of stack S into x else End.

Step 4 If ${}^\bullet x \ne \varnothing$ then for $\forall y \in {}^\bullet x$ do Step 5 else $P_{0s} \leftarrow P_{0s} \cup \{x\}$, goto Step 3;

Step 5 If $y \in P$ then $P_s \leftarrow P_s \cup \{y\}$ else $T_s \leftarrow T_s \cup \{y\}$;

Step 6 Set $F_s \leftarrow F_s \cup \{(y,x)\}$; Let y into stack S; goto Step 3.

Algorithm *REASONING* (Reasoning of Fuzzy Logical Petri Net)

Input: $SFLPN = (P_s, T_s; F_s, M_{0s}, D_s, h_s, \alpha_s, \theta_s, \lambda_s)$, $P_g \subseteq P_s$.

Output: $\lambda_s(p_g), \forall p_g \in P_g$.

Step 1 Let $\alpha_s(p) \leftarrow 0, \forall p \in P_s - P_{0s}$; $b(t) \leftarrow 0, \forall t \in T_s$

Step 2 Let each p_0 into stack S, for $\forall p_0 \in P_{0s}$

Step 3 If $S \neq \emptyset$ then let top element of stack S into x else End.

Step 4 If $x \in P_s$ then goto Step 9 else do Step 5

Step 5 If $x \in T_s^1$ then do Step 6 else goto Step 7

Step 6 If $\alpha_s(p_i) \geq \theta_s(x)$ then $\alpha_s(p_j) \leftarrow \max\{\alpha_s(p_j), \lambda_s(x)\alpha_s(p_i)\}$, goto Step 12

Step 7 If $x \in T_s^{AND}$ then do Step 8 else goto Step 9

Step 8 If $\min\limits_{p_i \in {}^\bullet x} \alpha_s(p_i) \geq \theta_s(x)$ then $\alpha_s(p_j) \leftarrow \max\{\alpha_s(p_j), \lambda_s(x)\min\limits_{p_i \in {}^\bullet x} \alpha_s(p_i)\}$, goto Step 12

Step 9 If $x \in T_s^{OR}$ then do Step 10 else goto Step 12

Step 10 If $\max\limits_{p_i \in {}^\bullet x} \alpha_s(p_i) \geq \theta_s(x)$ then $\alpha_s(p_j) \leftarrow \max\{\alpha_s(p_j), \lambda_s(x)\max\limits_{p_i \in {}^\bullet x} \alpha_s(p_i)\}$, goto Step 12

Step 11 If $(\alpha_s(p_i) \geq \theta_s(x)) \wedge (\alpha_s(p) = 0, p \in {}^\bullet x - \{p_i\})$ then $\alpha_s(p_j) \leftarrow \max\{\alpha_s(p_j), \lambda_s(x)\alpha_s(p_i)\}$, goto Step 12

Step 12 If $x^\bullet \neq \emptyset$ then for $\forall y \in x^\bullet$ if $(y \in T_s) \wedge (b(y) = 0)$ then let y into stack S

Step 13 Goto Step 3.

5. Application of Fuzzy Logical Petri Nets

Table 3 shows the set of rules for representing a partial fault model of a car engine described in [28]. The analysis of this problem proceeds according to our methods.

Table 3. Rules Set and Corresponding Transitions

No.	Rules	Transitions
1	IF pist-ring-state(worn) OR pist-state(worn) THEN oil-cons(incr)	t_1^{OR}
2	IF oil-cons(incr) THEN ex-smoke(black)	t_2
3	IF oil-cons(incr) THEN lack-of-oil(sev)	t_3
4	IF lack-of-oil(sev) THEN eng-temp(incr)	t_4
5	IF lack-of-oil(sev) THEN oil-light(red)	t_5
6	IF eng-temp(incr) THEN temp-ind(red)	t_6
7	IF eng-temp(incr) THEN accel-resp(del)	t_7
8	IF road-cond(poor) AND ground-clear(low) THEN oil-sump(holed)	t_8^{AND}
9	IF oil-sump(holed) THEN lack-of-oil(sev)	t_9
10	IF oil-sump(holed) THEN hole-oil-sump(yes)	t_{10}
11	IF spark-plug-mileage(high) THEN spark-plugs(used-up)	t_{11}
12	IF spark-plugs(used-up) THEN spark-ign(irr)	t_{12}
13	IF spark-ign(irr) THEN accel-resp(del)	t_{13}

Table 4. Propositions Set and Correspondence With Places Set

No.	Propositions	Places	No.	Propositions	Places
1	pist-ring-state(worn)	p_1	9	accel-resp(del)	p_9
2	pist-state(worn)	p_2	10	road-cond(poor)	p_{10}
3	oil-cons(incr)	p_3	11	ground-clear(low)	p_{11}
4	ex-smoke(black)	p_4	12	oil-sump(holed)	p_{12}
5	lack-of-oil(sev)	p_5	13	hole-oil-sump(yes)	p_{13}
6	eng-temp(incr)	p_6	14	spark-plug-mileage(high)	p_{14}
7	oil-light(red)	p_7	15	spark- plugs(used-up)	p_{15}
8	temp-ind(red)	p_8	16	spark-ign(irr)	p_{16}

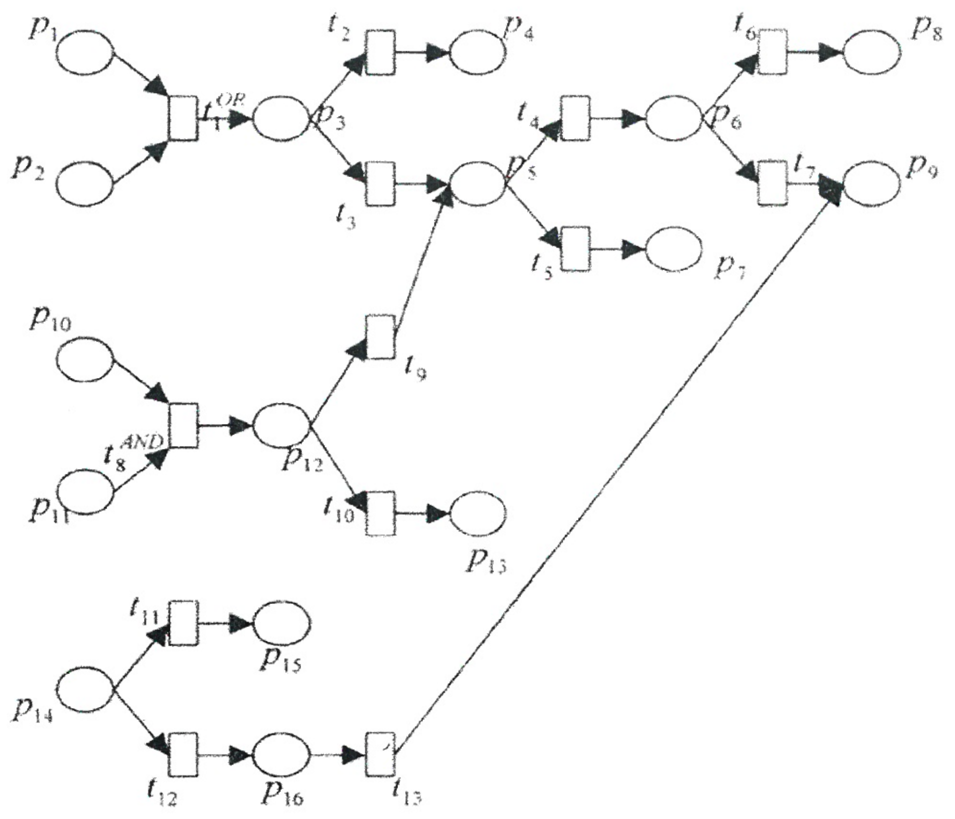


Figure 1. A Fuzzy Logical Petri Net ***LFPN*** $_1$

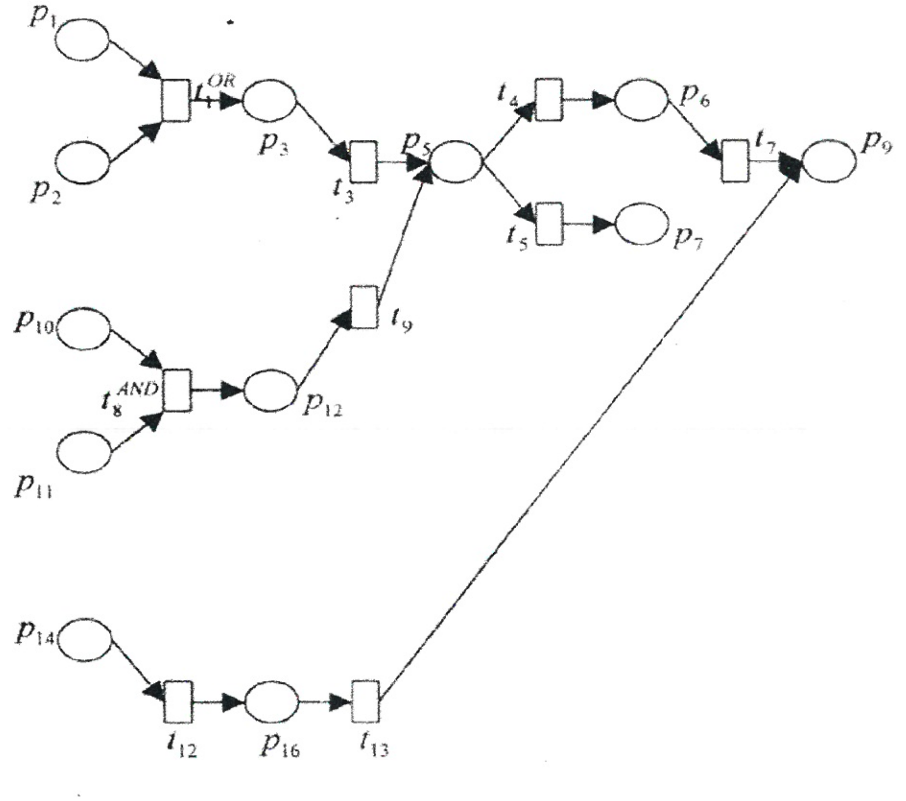


Figure 2. A Sub-net ***SLFPN*** $_1$ **of the Fuzzy Logical Petri Net** ***LFPN*** $_1$

Table 5. Given Values of Fuzzy Rules

t	t_1^{OR}	t_8^{AND}	t_{12}	t_3	t_9	t_{13}	t_4	t_5	t_7
$\lambda(t)$	0.82	0.67	0.75	0.77	0.95	0.88	0.83	0.71	0.86
$\theta(t)$	0.34	0.55	0.29	0.51	0.45	0.56	0.42	0.39	0.40

Table 6. The Computing Results of Fuzzy Reasoning Based on Fuzzy Logical Petri Nets

p_s	p_1	p_2	p_{10}	p_{11}	p_{14}	p_3	p_{12}	p_{16}	p_5	p_6	p_7	p_9
$\alpha(p_s)$	0.92	0.85	0.88	0.79	0.82	0.75	0.53	0.58	0.62	0.48	0.41	0.55

6. Conclusion

This paper proposed that a new representational model for a kind of fuzzy production systems is established and the reasoning algorithms are presented based on Petri net. It is semantics which is mainly considered in this model. The algorithms procedures proposed in this paper are based on the results of the reflection application on the input methodology [15] to several cases of fuzzy inconsistencies as analysed in [21]. The application of this model and the reasoning algorithm are discussed.

REFERENCES

1. AGARWAL, R. and TANNIRU, M., **A Petri Net Based Approach for Verifying the Integrity of Production Systems**, INT. J. MAN-MACHINE STUDIES, Vol. 36, 1992, pp. 447-468.
2. GIORDANA, A. and SAITTA, L., **Modeling Production Rules By Means of Predicate Transition Networks**, INFORMATION SCIENCES, Vol. 35, No. 1, 1985, pp. 1-41.
3. PETERKA, G. and MURATA, T., **Proof Procedure and Answer Extraction in Petri Net Model of Logical Programs,** IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, Vol. 15, No. 1, 1989, pp. 209-217.
4. CHOW, T.W.S., and LI, J.Y., **Higher-order Petri Net Models Based on Artificial Neural Networks**, Vol. 92, No. 1, 1997, pp. 289-300.
5. PORTINALE, L., **Exploiting T-invariant Analysis in Diagnostic Reasoning on A Petri Net Model**, LNCS 691, SPRINGER-VERLAG, 1993, pp. 339-356.
6. PORTINALE, L., **Verification of Causal Models Using Petri Nets**, INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, Vol. 7, No. 8, 1992, pp. 715-742.
7. DENG, Y. and CHANG, S.K., **A G-Net Model for Knowledge Representation and Reasoning**, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, Vol. 2, No. 3, 1990, pp. 295-309.
8. LOONEY C.G., **Fuzzy Petri Nets for Rule-based Decision Marking**, IEEE TRANSACTIONS ON SYSTEM, MAN AND CYBERNETICS, Vol. 18, No. 1, 1988, pp. 178-183.
9. CHEN, S., KE, J. S. and CHANG, J., **Knowledge Representation Using Fuzzy Petri Nets**, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, Vol. 2, No. 3, 1990, pp. 311-319.
10. CHANG, A. and HALL, L., **The Validation of Fuzzy Knowledge Based Systems**, in L. Zadeh and J. Kacprzyk, (Eds.) Fuzzy Logic and the Management of Uncertainty, JOHN WILEY & SONS, 1992, pp. 589-604.
11. GUPTA, M. M. and QI, J., **Theory of T-norms and Fuzzy Inference Methods**, FUZZY SETS AND SYSTEMS, 40, 1991, pp. 431-450.

12. MIZUMOTO, M. and ZIMMERMANN, H., **Comparison of Fuzzy Reasoning Methods**, FUZZY SETS AND SYSTEMS, 8, 1982, pp. 253-283.

13. YAGER, R. and LARSEN, H., **On Discovering Potential Inconsistencies in Validating Uncertain Knowledge Bases by Reflecting on the Input**, IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS, Vol. 21, No. 4, 1991, pp. 790-801.

14. KONAR, A. and MANDAL, A. K., **Uncertainty Management in Expert Systems Using Fuzzy Petri Nets**, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, Vol. 8, No. 1, 1996, pp. 96-105.

15. GARG, M. L., ASHON, S. I. and GUPTA, P. V., **A Fuzzy Petri Net for Knowledge Representation and Reasoning**, INF. PROCESS. LETT., 39, 1991, pp. 165-171.

16. BUGARIN, A. J. and BAROO, S., **Fuzzy Reasoning Supported by Petri Nets**, IEEE TRANSACTIONS ON FUZZY SYSTEMS, Vol. 2, No. 2, 1994, pp. 135-150.

17. ZHOU, M., MANIKOPOULOS, C. N. and NERURKAR, S. S., **Design and Implementation of Fuzzy Logical Embedded Programmable Logical Controllers for Process Control**, Conference on Computer Integrated Manufacturing Process Industries, 1992, pp. 703-717.

18. MURATA, T., **Temporal Uncertainty and Fuzzy-Timing High-Level Petri Nets,** LECTURE NOTES IN COMPUTER SCIENCE, Vol. 1091, 1996, pp. 29-58.

19. ANDREU, D., PASCAL, J. C. and VALETTE, R., **Fuzzy Petri Net-Based Programmable Logical Controller,** IEEE TRANSACTIONS ON SYSTEM, MAN AND CYBERNETICS, Vol. 27, No. 6, 1997, pp. 952-961.

20. PEDR`'CZ, W. and GOMIDE, F., **A Generalized Fuzzy Petri Net Model**, IEEE TRANSACTIONS ON FUZZY SYSTEMS, 2, 4, 1994, pp. 295-301.

21. SCARPELLI, H. and GOMIDE, F., **Fuzzy Reasoning and High Level Fuzzy Petri Nets**, Proceedings of the 1st European Congress on Fuzzy Intelligent Technology, Aachen, Germany, 1993, pp. 600-605.

22. SCARPELLI, H., GOMIDE, F. and PEDYCZ, W., **A High Level Fuzzy Petri Net Approach for Discovering Potential Inconsistencies in Fuzzy Knowledge Bases**, FUZZY SETS SYSTEMS, Vol. 19, 1994, pp. 175-193.

23. GENRICH, H.J., **Predicate/Transition Nets,** in W.Brauer , W. Reisig, and G. Rozenberg (Eds.) Petri Nets: Central Models and Their Properties, SPRINGER-VERLAG, New York, 1986, pp. 207-247.

24. JENSEN, K., **Colored Petri Nets**, in W. Brauer , W. Reisig, and G. Rozenberg (Eds.) Petri Nets: Central Models and Their Properties, SPRINGER-VERLAG, New York, 1986, pp. 248-299.

25. SCARPELLI, H., GOMIDE, F. and PEDYCZ, W., **A Reasoning Algorithm for High Level Fuzzy Petri Nets**, IEEE TRANSACTIONS ON FUZZY SYSTEMS, Vol. 4, No. 3, 1996, pp. 282-293.

26. PAL, S. and KONAR, A., **Cognitive Reasoning Using Fuzzy Neural Nets**, IEEE Transactions On System, Man and Cybernetics, 26, 4, 1996.

27. LEUNG, K. S. and SO, Y. T., **Inconsistency in Fuzzy Rule-based Expert Systems**, Proceedings of the International Conference on Fuzzy Logic and Neural Networks, Japan, 1990, pp. 849- 852.

28. LEE, J., LIU, K. and CHIANG, W., **Fuzzy Petri Nets for Modeling Rule-Based Reasoning**, INTERNATIONAL JOURNAL ON ARTIFICIAL INTELLIGENCE, Vol. 7, No. 4, 1998, pp. 463-485.

29. PORTINALE, L., **Exploiting T-invariant Analysis in Diagnostic Reasoning on A Petri Net Model**, LNCS, Vol. 691, SPRINGER-VERLAG, 1993, pp. 339-356.

30. MURATA, T., **Petri Nets: Properties, Analysis and Applications**, Proceedings IEEE, 77, 4, 1989, pp. 541-580.

31. .JIANG, C. J. and WU, Z.H., **Net Operations (I),** J. OF COMP. SCI. AND TECH., Vol.7, No.4, 1992, pp.333-344.

32. JIANG, C.J., **Net Operations (II),** J. OF COMP. SCI. AND TECH., Vol.10, No.6, 1995, pp.509-517.