Fifth IFSA World Congress (1993), 1326-1329

FUZZY REASONING AND FUZZY PETRI NETS

HELOISA SCARPELLI* and FERNANDO GOMIDE+

*UFSCar/CCT/DC - Cx. Postal 676 CEP 13565 -905 - São Carlos - SP - Brazil
+Unicamp/FEE/DCA - Cx. Postal 6101 CEP 13081-970 - Campinas - SP - Brazil

This work presents a net-based structure to model approximate reasoning using fuzzy production rules, the Fuzzy Petri Net model. The Fuzzy Petri Net model is formally defined as a n-uple of elements. It allows for the representation of simple and complex forms of rules such as rules with conjunction in the antecedent and qualified rules. Parallel rules and conflicting rules can be modeled as well. We also developed an analysis method based on state equations and two fuzzy reasoning algorithms. Finally, the proposed method is applied to an example.

1 Introduction

A fuzzy production rule describes the fuzzy relation between two fuzzy propositions([4],[1]). Fuzzy propositions are assertions that the value of a variable is one particular fuzzy subset in the domain of that variable. For example, the value *high* in the proposition *The temperature of the patient is high*, can be represented as a fuzzy subset A of interval of temperatures [0,50]. Stated more formally, assume V and U are variables that take their values in the base sets X and Y, respectively. The fuzzy production rule has the form

$$\text{IF } V \text{ is } A \text{ THEN } U \text{ is } B,$$

where A and B are two fuzzy subsets of base sets X and Y.

The most common inference pattern in approximate reasoning, called the compositional rule of inference, has the form:

$$H_1 \;:\; V \textit{ is } A' \quad (1)$$

$$H_2 \;:\; \underline{\text{IF } V \text{ is } A \text{ THEN } U \text{ is } B} \quad (2)$$

$$U \textit{ is } B' \quad (3)$$

where A' and B' are approximations of A and B respectively.

The essential laws to infer further data in this approach are based upon the theory of approximate reasoning([7]), according to which the conditional statement (2) induces a fuzzy relation D on $X \times Y$. A number of possible forms exist for the representation of D. In [2], a generalized fuzzy reasoning method is proposed by extending T-operators[1] to conventional fuzzy reasoning methods in which *min* and *max* operators are widely being used. A general representation of implication functions is defined by $D_{A\to B}(x,y) = f_{\to}(A(x), B(y))$.

In order to make inferences with fuzzy concepts,the consequent B' in (3) is calculated as $B' = A' \circ D$ where $B'(y) = sup_x \star(A'(x), D_{A\to B}(x,y))$.

We shall refer to cases where the variables range over finite sets.

2 The Fuzzy Petri Net Model

A Fuzzy Petri net(FPN) is a bipartite graph which contains two types of nodes: places and transitions, where circles represent places and bars represent transitions. The relationships from places to transitions and from transitions to places are represented by directed arcs. Each arc

[1] In what follows, we shall denote the triangular norm as $\star$ and triangular conorms as s.

is associated with a weight value between zero and one. Arcs leaving a place may be attached to each other by a transversal arc, and are called *and arcs.* The concept of FPN([5]) introduced here is derived from Petri nets([3]) and from a different Fuzzy Petri Net defined in [1]. A generalized Fuzzy Petri Net structure can be defined as a 8-tuple $FPN = (P, T, S, F, w, \alpha, \beta, \Gamma)$, where

$P = \{p_1, p_2, \cdots, p_n\}$ is a finite set of places,

$T = \{t_1, t_2, \cdots, t_m\}$ is a finite set of transitions,

$P \cap T = \emptyset$, P and T are partitioned into disjoint subsets called *groups*,

$S = X^1 \cup X^2 \cup \cdots \cup X^k$ is a union of finite base sets X^i such that $X^i = \{x_1^i, x_2^i, \cdots, x_l^i\}$, $i = 1, \cdots, k$,

$F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (flow relation),

$\omega : F \rightarrow [0,1]$ is a weight function, a mapping from arcs to real values between zero and one,

$\alpha : P \rightarrow [0,1]$ is a mapping from places to real values between zero and one,

$\beta : P \rightarrow D$ is a bijective mapping from places to elements of base sets,

Γ is a triple (γ, δ, τ) such that γ and δ are T-norm operators and τ is a T-conorm operator.

A FPN with some places containing tokens is called a *marked fuzzy Petri net.* The token in a place p_i is represented by a labeled dot $\overset{\alpha(p_i)}{\bullet}$ and α is called a *fuzzy marking* of the FPN.

The inference pattern above can be represented by the marked FPN in figure 1(a), where $\alpha(p_i) = A'(x_i) = a_i'$, $x_i \in X, i = 1, \cdots, n$, A and B are fuzzy subsets of the base sets $X = \{x_1, x_2, \cdots, x_n\}$ and $Y = \{y_1, y_2, \cdots, y_n\}$ respectively, and $D_{A \rightarrow B}$ is given by D_{ij}. The set of places $P = \{p_1, p_2, \cdots, p_{n+m}\}$ is partitioned in $P^1 = \{p_1, \cdots, p_n\}$ and $P^2 = \{p_{n+1}, \cdots, p_{n+m}\}$, the set of trasitions $T = T^1 = \{t_1, \cdots, t_m\}$, and the arc weights are $\omega(< p_i, t_j >) = d_{ij}$, $\omega(< t_j, p_{n+j} >) = 1$, $i = 1, \cdots, n$, $j = 1, \cdots, m$.

The behavior of a fuzzy rule based system can be simulated by changing the state or marking in a FPN, according to the transition rules. We define, for each transition $t \in T$, the input set $I(t)$ and the output set $O(t)$ as $I(t) = \{p \in P \mid (p,t) \in F\}$, $O(t) = \{p \in P \mid (t,p) \in F\}$. In a FPN, a transition t_i is enabled to fire if all $p_j \in I(t_i)$ has a token in it. Each transition in a fuzzy Petri net can be of one of four types, depending on the operation applied to find the new token value. Transitions belonging to the same group are of the same type and must fire all at once. A firing of an enabled transition t removes $1/n$ token from each input place p of t and add one token to each of its output places, where $n = \mid O(p) \mid$.

Let t_j be a transition, $O(t_j) = p_r$ and $I(t_j) = \{p_1, p_2, \cdots, p_n\}$. The token value in p_r after t_j fires, is given by:

1. If t_j is a type 1 transition, then

   $$\alpha(p_r) = b_j' = sup_i(a_i' \gamma d_{ij})$$

   where d_{ij} is the weight of the arc connecting place p_i to transition t_j.

2. If t_i is a type 2 transition, then

   $$\alpha(p_r) = \delta_{i=1,\cdots,n}(a_{k_i}^i).$$

3. If t_i is a type 3 transition, then

   $$\alpha(p_r) = b_j' = \alpha(p)$$

   where $I(t_j) = \{p\}$.

4. If t_4 is a type 4 transition, then

   $$\alpha(p_r) = b_j' = \tau_{i} = 1, \cdots, n(a_{k_i}^i).$$

The marked net before and after transitions firing for the conditional rule (2) is illustrated in figure 1(a) and (b). All transitions are of type 1.

More complex forms of rules can easily be represented in this approach:

- *Rules with multiple antecedent* - Composite rules of the form IF V_1 is A_1 and V_2 is A_2 and $\cdots$ and V_n is A_n THEN U is B shall be modeled in the FPN using an extra group of places and one of type 2 transitions in an intermediate level, which combine the sets in the antecedent.

- *Qualified rules* - rules with quantifiers in the antecedent and rules with certainty qualification can be modeled when interpreted under Yager's methodology ([6]). Concerning to the FPN model, the same basic structure used for simple rules can be used to model qualified rules, the only difference lying in the evaluation of arc labels.

- *Paralell rules* - rules of the form IF V is A_i THEN U is B_i for $i = 1, \cdots, n$ are called *parallel rules.* The

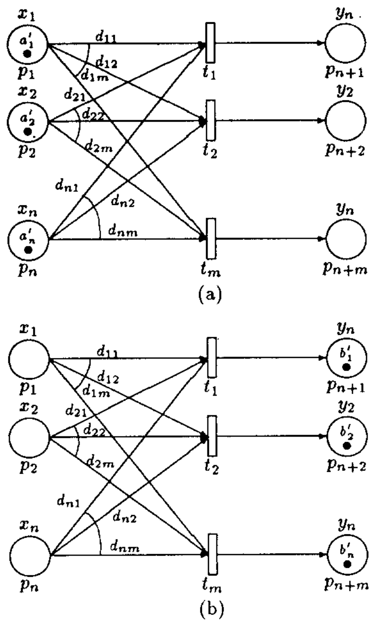

Figure 1: Firing a marked fuzzy Petri net. (a) Before firing transitions $t_1, t_2, \cdots, t_m$. (b) After firing transitions $t_1, t_2, \cdots, t_m$.

overall fuzzy relation D is then given by

$$D(x,y) = \mathrm{s}_{i=1}^{n}(D_{A_i \to B_i}(x,y)).$$

Given A', the consequent B' is calculated as $B' = A' \circ D$, where

$$B'(y) = sup_x \star (A'(x), \mathrm{s}_{i=1}^{n}(f_{\to}(A_i(x), B_i(y)))).$$

The basic Fuzzy Petri Net for this set of parallel rules has the same structure as a single rule, the only difference being the value of arc weights. An alternative way for implementing parallel rules is to perform each inference $A', A_i \to B_i \vdash B_i'$ separately and then combine the (B_i')'s. In the second case, we need the type 3 transition, for making tokens copies, so that all the parallel rules have their corresponding transition groups enabled to fire and the type 4 transition, that applies the aggregation operator on the sets B_i', $i = 1, \cdots, n$, resulting from each individual inference.

- *Conflicting rules* - these are the rules possessing the same variables in the antecedent but different variables in the consequent, giving rise to independent reasoning paths. We model this type of rules designing a single place group for the antecedent of both rules, one place group for each of the consequents and a transition group for each rule. The antecedent place group is connected to each transition group by a set of and arcs. The tokens in the antecedent place groups propagate their values through only one of the and arc sets.

Besides describing a rule-based system, FPN are a usefull tool to analyse the modeled system . We have modified the well known analysis method of Matrix Equation for ordinary Petri nets so that it can be applied to FPN. The modified method represents only token positions and their changes and not token values. The basic concepts of *Incidence Matrix*, *State Equation* and *Necessary Reachability Condition* have been redefined and can be applied when investigating the existence of relations between propositions ([5]).

3 Fuzzy Reasoning Algorithms

Two types of fuzzy reasoning algorithms based on the FPN model have been defined:

1. *Foward Chaining Algorithm* - the net is initially marked with tokens representing facts in the knowledge base. The places with an initial token are called *starting places*. The algorithm automatically generates all the reasoning paths from the starting places, until there is no more transitions to fire.
2. *Backward Chaining Algorithm* - the second type of reasoning algorithm is performed in two phases: first, a knowledge tree is built, starting from the goal as the root and including all the subgoals that have to be solved in the reasoning process. In a second phase, the tree is searched from the leaves to the root and the fuzzy sets are calculated.

4 Application Example

As an application example, we have considered a knowledge based system consisting of fuzzy production rules with the objective of analysing the behavior of a flexible machining cell that consists of a number of machining centers. This example has been usefull in describing the situations where the design of the FPN model by combining the basic structures requires special attention. The knowledge base consists of a set of rules relating the variables involved in the problem in terms of specific linguistic values, such as *IF fabrication batch size is small and transfer batch size is medium THEN end time is small*, or *IF transfer batch size is large THEN utilization of CN-72 is small.*

Let us suppose that we know the facts *The fabrication batch size is around 60* and *The transfer batch size is around 15.* After performing the algorithm, the final net state is interpreted as *The end time is almost medium*, *The CN-72 utilization is small* and *The medium queue size is not very small and not very medium.*

5 Conclusions

We have proposed a very general and powerfull net-based tool to model fuzzy reasoning, where different kinds of fuzzy reasoning methods can be represented. The FPN model can be used when the knowledge base is composed of a great variety of simple and complex rules, like multiple antecedent rules, qualified rules, parallel rules and conflicting rules. The approach introduced here has been shown to be usefull for modeling, designing, verifying and implementing knowledge bases and fuzzy systems.

6 References

1. CHEN,S., KE,J. S. and CHANG,J. - *Knowledge Representation using Fuzzy Petri Nets*, IEEE Trans. Knowledge nad Data Engineering, 2, 3, 1990, pp. 311-319.

2. GUPTA, M. M. and QI, J. - *Theory of T-norms and fuzzy inference methods*, Fuzzy Sets and Systems, 40, 1991, pp. 431-450.

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

4. NEGOITA, C. V. - *Expert Systems and Fuzzy Systems*, Benjamin/ Cummings, 1985.

5. SCARPELLI,H. and GOMIDE, F. - *Modeling Fuzzy Reasoning using Fuzzy Petri Nets*, Technical Report RT-DCA 020/92, DCA/FEE/UNICAMP, 1992.

6. YAGER, R. R. - *Approximate Reasoning as a Basis for Rule-Based Expert Systems*, IEEE Trans. on SMC, 14, 4, 1984, pp.1028-1034.

7. ZADEH,L. A. - *A Theory of Approximate Reasoning.* Machine Inteligence 9, Hayes, Michie & Kulich, (eds.), 9, 1979, pp.149-194.