Journal of Computing and Information Technology - CIT 2, 1994, 3, 181–191

Fuzzy Expert System for Pattern Recognition*

Mojca Parkelj and Nikola Pavešić

Faculty for Electrical Engineering and Computer Science, University of Ljubljana, Ljubljana, Slovenia

In a pattern recognition environment it is often difficult to capture information in a precise form, and consequently, the knowledge about the recognition domain (problem) may be inexact, incomplete or not reliable. In order to properly represent the knowledge of this kind, fuzzy production rules have been used. In this paper fuzzy Petri nets are used as a knowledge representation scheme in which the fuzzy production rules are applied for describing a fuzzy relation between two propositions. Based on the fuzzy Petri net model of a knowledge base, an efficient algorithm is proposed to perform fuzzy reasoning automatically.

1. Introduction

Information about a pattern recognition domain (problem) can be uncertain, fuzzy, uncertain and fuzzy and/or fuzzy uncertain.

Uncertainty occurs when one is not absolutely certain about a piece of information. The degree of uncertainty is usually represented by a numerical value between zero and one. It represents a strength of a belief in the fact or in the rule and it is named *certainty factor (CF)*. For example:

X is *a bird*. (CF=0.8)
IF (*X* is *a bird*) THEN (*X* can *fly*). (CF=0.9)

The certainty factors are 0.8 and 0.9.

Fuzziness occurs when the boundary of a piece of information is not clear-cut. It is represented by *fuzzy terms*. For example:

The cube is *small*.
IF (*the price* is *high*) THEN (*the profit* is *good*).

Small, high and *good* are fuzzy terms.

Uncertainty and fuzziness may occur simultaneously. For example:

The cube is *small*. (CF=0.8)
IF (*the price* is *high*) THEN (*the profit* is *good*). (CF=0.9)

Sometimes the uncertainty can also be fuzzy. The fuzzy uncertainty is modeled by *fuzzy numbers*. For example:

The cube is *small*. (around 0.8)

Here, *around 0.8* is the fuzzy uncertainty and *small* is a fuzzy term.

This article presents a knowledge based system that allows any mix of fuzzy and normal terms as well as uncertainties in the rules and facts. To provide this task, it employs fuzzy logic to handle inexact reasoning and certainty factors to handle the uncertainty.

In the Section 2. we deal with the representation of a fuzzy knowledge base in the Petri net formalism. The idea of using Petri net theory for knowledge representation was carried out in 1978 by S. Ribarić [10]. Later (1990), S. M. Chen et al. in [4]proposedfuzzy Petri nets as a knowledge representation scheme in which the fuzzy production rules are applied for describing the fuzzy relation between two propositions.

The inference engine is described in the Section 3. Treatments of fuzzy sets, fuzzy logic and fuzzy models can be found in [1, 5, 13]and [14],

* We gratefully acknowledge the support of the Ministry of Science and Technology of Slovenia and Alexander von Humboldt Foundation.

the representation of a fuzzy knowledge base in the Petri nets formalism in [4, 8, 10, 11, 12], whereas the rule evaluation and fuzzy reasoning algorithm are explained in [2, 3, 4, 15]. The fuzzy reasoning algorithm proposed in [4]was based on the certainty factor approach. The fuzzy reasoning algorithm proposed in this paper is based on the certainty factor approach as well as on the fuzzy logic approach.

2. Fuzzy knowledge base

In the fuzzy knowledge base the knowledge entities, such as facts and production rules are stored. These knowledge entities provide information that enables the inference engine to perform consultations.

A fact is a data proposition of the form:

OBJECT is VALUE
(fuzzy/nonfuzzy uncertainty)

where an object is the basic entity in the system. It is uniquely identified by the object name and the attribute(s). The attribuite(s) can be empty if an object name is sufficient to describe the object. The object can be fuzzy or not. The values of a nonfuzzy object are numbers or literal strings, but if the object is fuzzy, its values are fuzzy terms. Fuzzy terms are represented by fuzzy sets and fuzzy sets are defined by their membership function. The uncertainty of the fact can be fuzzy or nonfuzzy. Fuzzy uncertainty is modeled by fuzzy numbers. A fuzzy number is actually a real-number fuzzy set that is both convex and normal. Nonfuzzy uncertainty is expressed as ordinary certainty factors. A certainty factor represents a degree of truth of the fact and its value 1 means that the fact is absolutely certain.

A production rule is defined as an implication statement expressing the relationship between a set of antecedent propositions and a set of consequent propositions. Attached to each rule is a fuzzy/nonfuzzy uncertainty describing the degree of confidence in the rule:

IF (A is V) THEN (C is U) (CF=μ)

where

- A is a antecedent object and C is the consequent object of the rule. They can be fuzzy or nonfuzzy.
- V and U are object's values.
- μ is a value of the certainty factor CF and represents the strength of the belief in the rule. The larger the value, the more the rule is believed in.

The antecedent part of a rule consists of a single propositions or any combination of two or more propositions connected by a logical AND or a logical OR. But the consequent part of a rule can contain only a single proposition or multiple propositions with AND conjunctions between them.

The most distinctive feature of a fuzzy knowledge base is that, beside facts and rules, it also stores fuzzy sets representing fuzzy terms.

We can use **a Fuzzy Petri net** to model the fuzzy production rules in the knowledge base. A fuzzy Petri net is a bipartite directed 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. A generalized fuzzy Petri net structure can be defined as an 8-tuple:

$$\textbf{Fuzzy Petri net} = (P, T, D, I, O, f, \alpha, \beta)$$

where

- $P = \{p_1, p_2, ...p_n\}$ is a finite set of places,
- $T = \{t_1, t_2, ...t_m\}$ is a finite set of transitions,
- $D = \{d_1, d_2, ...d_n\}$ is a finite set of propositions,
- I: $T \rightarrow P^\infty$ is the input function - a mapping from transitions to bags of places,
- O: $T \rightarrow P^\infty$ is the output function - a mapping from transitions to bags of places,
- f: $T \rightarrow [0, 1]$ is an association function - a mapping from transitions to real values between zero and one,
- α: $P \rightarrow [0, 1]$ is an association function - a mapping from places to real values between zero and one,
- β: $P \rightarrow$ D is an association function - a bijective mapping from places to propositions.

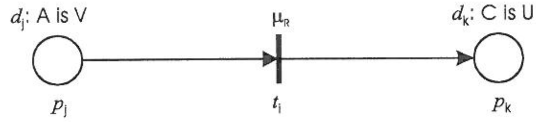

Fig. 1. A fuzzy Petri net.

Let A be a set of directed arcs. If $p_j \in I(t_i)$, then there exists a directed arc a_{ij}, $a_{ij} \in A$, from the place p_j to the transition t_i. If $p_k \in O(t_i)$, then there exists a directed arc a_{ik}, $a_{ik} \in A$, from the transition t_i to the place p_k.

As an example, by using a fuzzy Petri net the simple fuzzy production rule of type:

IF (d_j: A is V) THEN (d_k: C is U) (CF=μ_R)

can be modeled as shown in Figure 1.

If the antecedent part or consequence part of a fuzzy production rule contains AND or OR connectors, then it is called **a composite fuzzy production rule**. The composite fuzzy production rules can be distinguished into four rule-types and they can also be modeled by a fuzzy Petri nets as shown in figures below.

TYPE 1:

RULE: IF (d_{j1}: A_1 is V_1) AND ... AND (d_{jn}: A_n is V_n)
THEN (d_k: C is U) (CF=μ_R)

TYPE 2:

RULE: IF (d_{j1}: A_1 is V_1) OR ... OR (d_{jn}: A_n is V_n
THEN (d_k: C is U) (CF=μ_R)

TYPE 3:

RULE: IF (d_j: A is V)
THEN (d_{k1}: C_1 is U_1) AND ... AND (d_{kn}: C_n is U_n) (CF=μ_R)

TYPE 4:

RULE: IF (d_j: A is V)
THEN (d_{k1}: C_1 is U_1) OR ... OR (d_{kn}: C_n is U_n) (CF=μ_R)

Rules of this type are unsuitable for deducing control because they do not make specific implications. Therefore, we do not allow type 4 rule to appear in the knowledge base.

Example 1:

Let's assume that the knowledge base of a rule-based system is made up of the following fuzzy production rules:

R_1: **IF** (d_1: A is A_1) **THEN** (d_2: B is B_1) (μ_{R1}=0.85)
R_2: **IF** (d_2: B is B_1) **THEN** (d_3: C is C_1) (μ_{R2}=0.80)

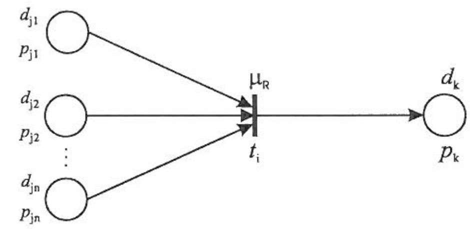

Fig. 2. Fuzzy Petri net representation of a type 1 rule.

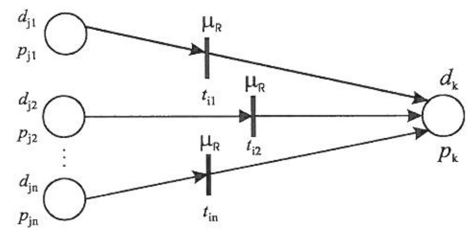

Fig. 3. Fuzzy Petri net representation of a type 2 rule.

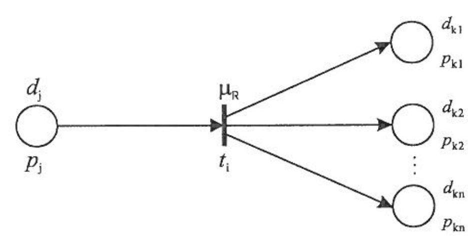

Fig. 4. Fuzzy Petri net representation of a type 3 rule.

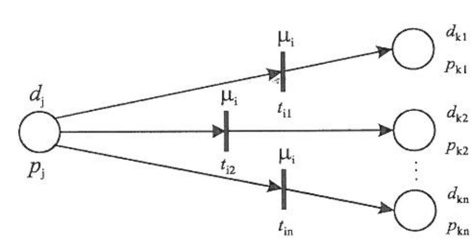

Fig. 5. Fuzzy Petri net representation of a type 4 rule.

R_3: **IF** (d_2: B is B_1) **THEN** (d_4: D is D_1) (μ_{R3}=0.80)
R_4: **IF** (d_4: D is D_1) **THEN** (d_5: E is E_1) (μ_{R4}=0.90)
R_5: **IF** (d_1: A is A_1) **THEN** (d_6: G is G_1) (μ_{R5}=0.90)
R_6: **IF** (d_6: G is G_1) **THEN** (d_4: D is D_1) **and** (d_9: J is J_1) (μ_{R6}=0.95)
R_7: **IF** (d_1: A is A_1) **and** (d_8: I is I_1) **THEN** (d_7: H is H_1) (μ_{R7}=0.90)
R_8: **IF** (d_7: H is H_1) **THEN** (d_4: D is D_1) (μ_{R8}=0.90)

The rules and the facts can be modeled with fuzzy Petri nets as shown in Figure 6.:

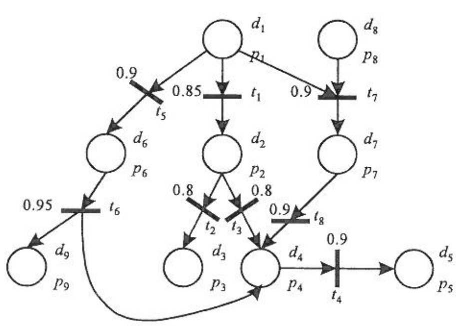

Fig. 6. Graphic representation of the knowledge base in the Petri net formalism.

3. Inference Engine

The function of the inference engine is to extract the knowledge stored in the fuzzy knowledge base and to make inferences from the respective rules and facts. It can evaluate simple and composite fuzzy production rules.

This Section is organised as follows. The subsection 3.1. describes evaluation of simple fuzzy production rule. Evaluation of composite fuzzy production rule is described in Subsection 3.2. The rule evaluation is a basic operation in a process of performing inferences from rules which make up the knowledge base. The subsection 3.3. presents the fuzzy reasoning process in the knowledge base which include rule chaining. The inference engine is making inferences by forward chaining.

3.1. Evaluation of a simple rule

Suppose there is a fuzzy production rule:

IF (A is V) THEN (C is U) (CF=μ_R)

where

- A is an antecedent object of the rule,
- C is a consequent object of the rule,
- μ_R is a certainty factor of the rule,
- V is a value of object A,
- U is a value of object C.

If an object is nonfuzzy, then its value is either a number or a string. If the object is fuzzy, its value is a fuzzy term. The object is treated as a linguistic variable. By a linguistic variable we mean a variable whose values are words or sentences in a natural or artificial language. Each value of a linguistic variable is represented by a fuzzy set. If an object A or C is fuzzy, then its value V or U is represented by fuzzy sets F_V or F_U, respectively.

We can say that a rule is fired by a fact and a conclusion is made simultaneously. The conclusion involves two information: an object value and a certainty factor of the conclusion.

Methods of simple rule evaluation:

I. Both objects A and C are nonfuzzy:

RULE: IF (A is V)
THEN (C is U) (CF=μ_R)

FACT: A is V (CF=μ_F)

CONCLUSION: C is U (CF=μ_C)

In order to apply this rule a fact object should be nonfuzzy. When the rule is evaluated, we obtain a conclusion object that is nonfuzzy too.

The certainty factor μ_C of the conclusion is calculated by multiplication of the certainty factor of the rule μ_R and the certainty factor of the fact μ_F. When certainty factors are fuzzy, then we should apply a fuzzy number multiplication.

$$\mu_C = \mu_R * \mu_F \tag{1}$$

II. Object A is nonfuzzy, object C is fuzzy:

RULE: IF (A is V) THEN
(C is U) (CF=μ_R)

FACT: A is V (CF=μ_F)

CONCLUSION: C is U (CF=μ_C)

A fact object should be nonfuzzy. A conclusion object obtained by the rule evaluation is fuzzy.

If C is a fuzzy object, then its value U is represented by fuzzy set F_U and the same fuzzy set F_U represents the value U in the conclusion.

The certainty factor μ_C of the conclusion is calculated by multiplication of the certainty factor of the rule μ_R and the certainty factor of the fact μ_F.

$$\mu_C = \mu_R * \mu_F \tag{2}$$

III. Both objects A and C are fuzzy:

RULE: IF (A is V) THEN (C is U) (CF=μ_R)

FACT: A is V' (CF=μ_F)

CONCLUSION: C is U' (CF=μ_C)

The fact object should be fuzzy. The conclusion object obtained by the rule evaluation is fuzzy too.

If both A and C are fuzzy objects, then the values V and U are represented by fuzzy sets F_V and F_U, respectively. We form a fuzzy relation R by performing some fuzzy operations of fuzzy sets F_V and F_U. There are different approaches to the fuzzy relation forming. For example, a fuzzy relation R can be formed by performing Cartesian product of fuzzy sets F_V and F_U.

Let F_V be a fuzzy set in the universe of discourse $\mathcal{V}$ and F_U be a fuzzy set in the universe of discourse $\mathcal{U}$. The Cartesian product of two fuzzy sets F_V and F_U is the fuzzy set of ordered pairs $(x,y), x \in \mathcal{V}, y \in \mathcal{U}$. The membership function of (x,y) in Cartesian product $F_V \times F_U$ is a fuzzy relation from $\mathcal{V}$ to $\mathcal{U}$ and is defined by:

$$m_R(x,y) = \min(m_{F_V}(x), m_{F_U}(y)) \tag{3}$$

where

- $m_R(x,y)$ is a membership function of the fuzzy relation R,
- $m_{F_V}(x)$ is a membership function of the fuzzy set F_V,
- $m_{F_U}(y)$ is a membership function of the fuzzy set F_U.

The value V' of the object A in the fact should be a fuzzy term represented by a fuzzy set $F_{V'}$. The fuzzy set $F_{U'}$ of the value U' in the conclusion is obtained by applying a fuzzy composition operation on $F_{V'}$ and R:

$$F_{U'} = F_{V'} \circ R \tag{4}$$

The membership function of the fuzzy set $F_{U'}$ is given by the max-min product of the membership functions $F_{V'}$ and R.

$$m_{F_{U'}}(y) = \max(\min(m_{F_{V'}}(x), m_R(x,y)) \tag{5}$$

The certainty factor μ_C of the conclusion is calculated by multiplication of the certainty factor of the rule μ_R and the certainty factor of the fact μ_F.

$$\mu_C = \mu_R * \mu_F \tag{6}$$

IV. Object A is fuzzy, object C is nonfuzzy:

RULE: IF (A is V) THEN (C is U) (CF=μ_R)

FACT: A is V' (CF=μ_F)

CONCLUSION: C is U (CF=μ_C)

If A is a fuzzy object, then its value V in the rule is represented by a fuzzy set F_V and its value V' in the fact is represented by fuzzy set $F_{V'}$. The conclusion is nonfuzzy. The certainty factor of the conclusion μ_C is obtained by multiplication of certainty factors μ_R, μ_F and **a similarity** M between F_V and $F_{V'}$, which are fuzzy sets of V and V', respectively. The similarity M measures how similar two fuzzy concepts represented by the two fuzzy sets are.

$$\mu_C = (\mu_R * \mu_F) \cdot M \tag{7}$$

The similarity M is calculated by the following algorithm:

IF $N(F_V; F_{V'}) > 0.5$
THEN M=$P(F_V; F_{V'})$
ELSE M=$(N(F_V; F_{V'})+0.5) \times P(F_V; F_{V'})$

where

- $P(F_V; F_{V'})$ is *a possibility* of a fuzzy data $F_{V'}$ given the fuzzy pattern F_V,
- $N(F_V; F_{V'})$ is *a necessity* of a fuzzy data $F_{V'}$ given the fuzzy pattern F_V.

The following are the formulas of the possibility and necessity measures between two fuzzy sets:

$$P(F_V;F_{V'}) = \max(\min(\mu_F(w), \mu_{F'}(w))) \quad (8)$$

$$N(F_V;F_{V'}) = 1 - P(F_V^C;F_{V'}) \quad (9)$$

where

- μ is a membership function of the above fuzzy sets,
- w is an element in the universe of discourse of the above fuzzy sets,
- F_V^C is a complement of F_V.

The possibility between two fuzzy sets gives the maximum of their intersection and measures to what extent they overlap. The necessity between two fuzzy sets reflects the following relationship between them:

$N(F_V;F_{V'}) > 0.5 \iff F_{V'}$ is *a concentration* of F_V

$N(F_V;F_{V'}) = 0.5 \iff F_{V'}$ is *a duplicate* of F_V

$N(F_V;F_{V'}) < 0.5 \iff F_{V'}$ is *a dilation* of F_V

If $F_{V'}$ is a concentration of F_V, it means that $F_{V'}$ has a more concentrated or narrower distribution than F_V. The more concetrated distribution of $F_{V'}$ represents a more strongly expressed term than that of F_V. For a dilation the situation is exactly the opposite.

In a fuzzy Petri net the evaluation of a fuzzy production rule can be considered as firing a corresponding transition. By using a fuzzy Petri nets, the fuzzy production rule and the fuzzy fact:

RULE: IF (d_j: A is V) THEN (d_k: C is U) (CF=μ_R)

FACT: d_f: A is V' (CF=μ_F)

can be modeled as shown in Figure 7.

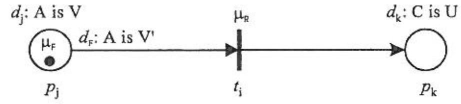

Fig. 7. Fuzzy production rule before firing transition

Let λ be *a threshold value*, where $\lambda \in [0, 1]$. Then:

- if $\mu_F \geq \lambda \Longrightarrow$ a rule may be evaluated,
- if $\mu_F < \lambda \Longrightarrow$ a rule cannot be evaluated.

A transition t_i is enabled to fire if the token values in all its input places are greater than the threshold value λ. A transition t_i fires by removing the tokens from its input places and then depositing one token into each of its output places. Simultaneously, the corresponding rule is evaluated. In the output place the conclusion is evaluated and the token value as certainty factor of the conclusion is calculated.

3.2. Evaluation of a composite rule

The composite production rules are distinguished into the following rule-types:

I. The antecedent part of the rule contains multiple propositions ($A_1, A_2, \ldots, A_n$) with AND connectors between them:

RULE: IF (A_1 is V_1) AND (A_2 is V_2) THEN (C is U) (CF=μ_R)

FACTS: A_1 is V_1' (CF=μ_{F_1})
A_2 is V_2' (CF=μ_{F_2})

CONCLUSION: C is U' (CF=μ_C)

If an object in the consequent proposition is nonfuzzy, no special treatment is needed. If the consequent proposition is fuzzy, the fuzzy set of the object value U' in the conclusion is calculated by considering *the distribution law* in logic:

$$(A_1 \text{ AND} A_2) \Rightarrow C$$
$$\equiv (A_1 \Rightarrow C) \text{ OR } (A_2 \Rightarrow C) \quad (10)$$

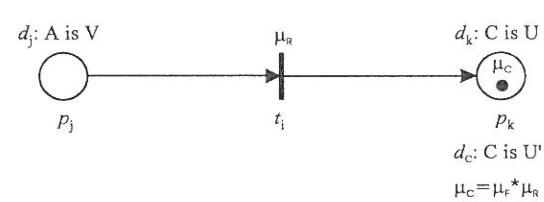

Fig. 8. Fuzzy production rule after firing transition

We can break up the composite fuzzy production rule into two simple fuzzy production rules and the fuzzy set $F_{U'}$ in the conclusion C is obtained by taking fuzzy union of the fuzzy sets F_{U_1} and F_{U_2}:

$$F'_U = F_{U_1} \cup F_{U_2} \tag{11}$$

where

- F_{U_1} is a fuzzy set in the conclusion C obtained from the first rule evaluation:

  RULE: IF (A_1 is V_1) THEN (C is U)
  FACT: A_1 is V'_1

- F_{U_2} is a fuzzy set in the conclusion C obtained from the second rule evaluation:

  RULE: IF (A_2 is V_2) THEN (C is U)
  FACT: A_2 is V'_2

  The certainty factor μ_C of the conclusion is calculated by the following formula:

$$\mu_C = \min(\mu_{F_1}, \mu_{F_2}) * \mu_R \tag{12}$$

II. The antecedent part of the rule contains multiple propositions ($A_1, A_2, \ldots, A_n$) with OR connectors between them:

RULE: IF (A_1 is V_1) OR (A_2 is V_2) THEN (C is U) (CF=μ_R)

FACTS: A_1 is V'_1 (CF=μ_{F_1})
A_2 is V'_2 (CF=μ_{F_2})

CONCLUSION: C is U' (CF=μ_C)

If an object in the consequent proposition is nonfuzzy, no special treatment is needed. If the consequent proposition is fuzzy, a fuzzy set of the object value U' in the conclusion is calculated by considering the distribution law in logic:

$$(A_1 \, \mathrm{OR} A_2) \Rightarrow C$$
$$\equiv (A_1 \Rightarrow C) AND (A_2 \Rightarrow C) \tag{13}$$

We can break up the composite fuzzy production rule into two simple fuzzy production rules and the fuzzy set $F_{U'}$ in the conclusion C is obtained by taking fuzzy intersection of the fuzzy sets F_{U_1} and F_{U_2}:

$$F'_U = F_{U_1} \cap F_{U_2} \tag{14}$$

where

- F_{U_1} is a fuzzy set in the conclusion C obtained from the first rule evaluation:

  RULE: IF (A_1 is V_1) THEN (C is U)
  FACT: A_1 is V'_1

- F_{U_2} is a fuzzy set in the conclusion C obtained from the second rule evaluation:

  RULE: IF (A_2 is V_2) THEN (C is U)
  FACT: A_2 is V'_2

  The certainty factor μ_C of the conclusion is calculated by the following formula:

$$\mu_C = \max(\mu_{F_1}, \mu_{F_2}) * \mu_R \tag{15}$$

III. The consequent part of the rule contains multiple propositions ($C_1, C_2, \ldots, C_n$) with AND connectors between them:

RULE: IF (A is V) THEN (C_1 is U_1) AND (C_2 is U_2) (CF=μ_R)

FACT: A is V' (CF=μ_F)

CONCLUSIONS: C_1 is U'_1 (CF=μ_{C_1})
C_2 is U'_2 (CF=μ_{C_2})

This rule can be decomposed into multiple rules with a single conclusion. We can break up the composite fuzzy production rule into two simple fuzzy production rules:

RULE1: IF (A is V) THEN (C_1 is U_1) (CF=μ_R)
RULE2: IF (A is V) THEN (C_2 is U_2) (CF=μ_R)

Each rule can be evaluated separately and all conclusions have the same certainty factors:

$$\mu_{C_1} = \mu_{C_2} = \mu_R * \mu_F \tag{16}$$

IV. The consequent part of the rule contains multiple propositions ($C_1, C_2, \ldots, C_n$) with OR connectors between them:

RULE: IF (A is V) THEN (C_1 is U_1) OR (C_2 is U_2) (CF=μ_R)
FACT: A is V' (CF=μ_F)

Rules of this type are unsuitable for deducing control because they do not make specific implications and we don't allow this type of rule to appear in the knowledge base.

Composite fuzzy production rule and their evaluation can also be modeled by a fuzzy Petri net as shown in figures below.

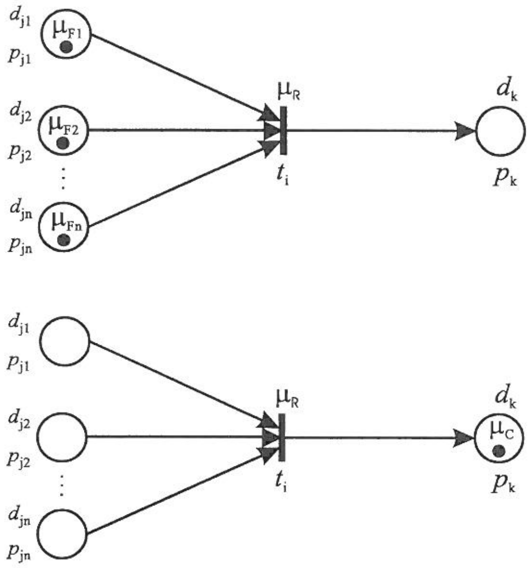

Fig. 9. Fuzzy Petri net representation of a type 1 rules.

TYPE 1:

RULE: IF (d_{j1}: A_1 is V_1) AND ... AND (d_{jn}: A_n is V_n) THEN (d_k: C is U)(CF=μ_R)

FACTS: A_1 is V_1' (CF=μ_{F_1})
...
A_n is V_n' (CF=μ_{F_n})

CONCLUSION: C is U' (CF=μ_C)

$$F_{U'} = F_{U_1} \cup F_{U_2} \cup ... \cup F_{U_n} \quad (17)$$

$$\mu_C = \min(\mu_{F_1}, ...\mu_{F_n}) * \mu_R \quad (18)$$

TYPE 2:

RULE: IF (d_{j1}: A_1 is V_1) OR ... OR (d_{jn}: A_n is V_n) THEN (d_k: C is U)(CF=μ_R)

FACTS: A_1 is V_1' (CF=μ_{F_1})
...
A_n is V_n' (CF=μ_{F_n})

CONCLUSION: C is U' (CF=μ_C)

$$F_{U'} = F_{U_1} \cap F_{U_2} \cap ... \cap F_{U_n} \quad (19)$$

$$\mu_C = \max(\mu_{F_1}, ...\mu_{F_n}) * \mu_R \quad (20)$$

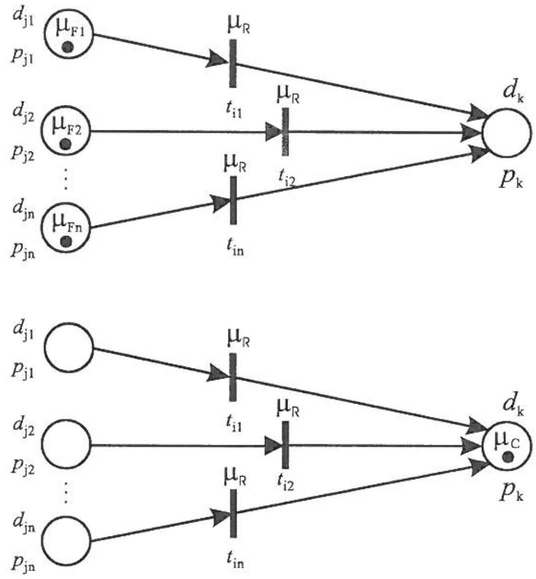

Fig. 10. Fuzzy Petri net representation of a type 2 rules.

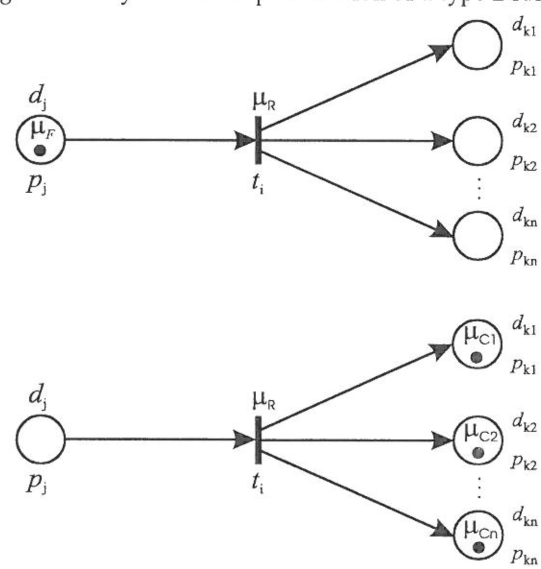

Fig. 11. Fuzzy Petri net representation of a type 3 rules.

TYPE 3:

RULE: IF (d_j: A is V) THEN (d_{k1}: C_1 is U_1) AND ... AND (d_{kn}: C_n is U_n) (CF=μ_R)

FACT: A is V' (CF=μ_F)

CONCLUSIONS: C_1 is U_1' (CF=μ_{C_1})
...
C_n is U_n' (CF=μ_{C_n})

$$\mu_{C_1} = ... = \mu_{C_n} = \mu_F) * \mu_R \quad (21)$$

3.3. Fuzzy reasoning supported by Petri nets

Let R be a set of fuzzy production rules:

$$R = \{R_1, R_2, ..., R_n\}$$

The general formulation of the i-th fuzzy production rule is as follows:

$$R_i\text{: IF } (d_A\text{: } A_i \text{ is } V_i) \text{ THEN } (d_C\text{: } C_i \text{ is } U_i)$$
$$(\text{CF}=\mu_{R_i})$$

In many situations, we may want to determine whether there exists an antecedent- consequence relationship from a proposition d_S to a proposition d_G. If a fact d_F for proposition d_S is given, we may want to ask what fuzzy set and what certainty factor of a conclusion might be evaluated. These problems can be solved by developing a fuzzy reasoning algorithm based on the fuzzy Petri net that was proposed by S. M. Chen in [4].

The propositions d_S and d_G are associated with the place p_S and p_G, respectively in the fuzzy Petri net. The places p_S and p_G are called **a starting place** and **a goal place**, respectively. The token value in the starting place p_S is certainty factor of the fact μ_F and the fact fuzzy set is also known.

All transitions that are derived from a starting place are enabled to fire if a token value in the starting place is greater than threshold value. A transition fires by removing the token from its input places and depositing one token into each of its output place. Corresponding rules are evaluated simultaneously. In the output places the token value is calculated and a conclusion is made. The evaluated conclusion behaves as a fact for the next transition firing.

The fuzzy reasoning algorithm proposed by S. M. Chen [4] can automatically generate all reasoning paths from a starting place to a goal place and if the fact in the starting place p_S is known, then the token value in the goal place p_G can be calculated and the conclusion is evaluated.

The algorithm can be expressed by a reasoning tree. Each node of the tree is denoted by a quadruplet:

$$(p_i, \alpha(p_i), F_{V_i'}, IRS(p_i))$$

where

- $p_i \in$ P is a place in fuzzy Petri net,
- $\alpha(p_i)$ is a token value in the place p_i,
- $F_{V_i'}$ is a fuzzy set of the fact in the place p_i,
- IRS(p_i) is a immediate reachability set of p_i.

A root node of the reasoning tree is denoted by the starting place p_S. The reasoning tree may have several terminal nodes denoted by a goal place p_G. They are marked as success nodes. If there are no success nodes, then there does not exist an antecedent-consequent relationship from a proposition d_S to a proposition d_G. A path from the root node to each success node is called **a reasoning path**. Along one reasoning path we can see in what order the rules have been evaluated from the starting proposition d_S to the goal proposition d_G.

To illustrate the fuzzy reasoning process an example is used.

Example 2:

The knowledge base of a rule-based system contains the fuzzy production rules as in Example 1. Assume that the threshold value is λ=0.2.

The two facts are given by the user:

$$F_1\text{: } (d_1'\text{: } A \text{ is } A_1') \ (\mu_{F_1}=0.80)$$
$$F_2\text{: } (d_8'\text{: } I \text{ is } I_1') \ (\mu_{F_1}=0.70)$$

We have to determine whether there exists an antecedent-consequence relationship from the proposition d_1 to the proposition d_4. The rules and the facts can be modeled by the fuzzy Petri net model as shown in Figure 12.

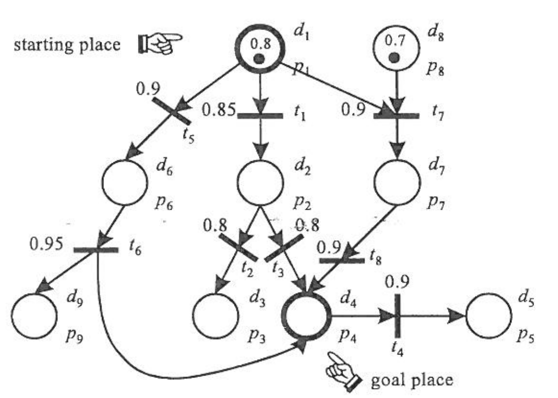

Fig. 12. Fuzzy Petri net representation of the knowledge base.

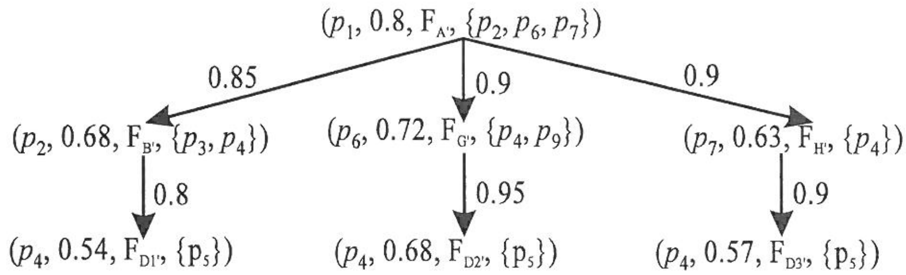

Fig. 13. Fuzzy reasoning tree.

The places p_1 and p_4 are called the starting place and the goal place, respectively.

After performing the fuzzy reasoning algorithm the fuzzy reasoning tree sprouts as shown in Figure 13.

There are three success nodes. Let Q be a set of success nodes:

$$Q = \{(p_4, 0.54, F_{D1'}, \{p_5\}), (p_4, 0.68, F_{D2'}, \{p_5\}), (p_4, 0.57, F_{D3'}, \{p_5\})\}$$

The dominant conclusion is that with the greatest certainty factor. Therefore, the fuzzy set of the conclusion is $F_{D2'}$ and the certainty factor of the conclusion is 0.68.

4. Conclusions

To facilitate better knowledge engineering and simulation of human reasoning, we have built into an expert system fuzzy concepts and inexact reasoning. The proposed expert system can handle both fuzziness and uncertainty, the two basic inexact concepts. The presented model of the fuzzy knowledge base based on fuzzy Petri nets has an ability to represent knowledge in a domain of application and supports fuzzy reasoning. The main properties of the presented fuzzy reasoning algorithm are:

- the reasoning tree of a fuzzy Petri net is finite,
- generating only necessary reasoning paths from an starting place to an goal place,
- an upper bound of the time complexity of the fuzzy reasoning algorithm is $O(nm)$, where n is a number of places, and m a number of transitions,
- an execution time of the fuzzy reasoning algorithm is proportional to a number of nodes in the reasoning tree generated by the algorithm.

Beside the fuzzy knowledge base and inference engine, the knowledge acquisition subsystem and user interface can be added to the fuzzy expert system. The knowledge acquisition subsystem is responsible for acquiring and managing rules and facts. The user interface caters for communications between a user and the system and comprehends linguistic approximation routine. An linguistic approximation routine is a process that maps a set of fuzzy sets onto a set of linguistic values or expressions. This process is needed for two purposes. One is to find corresponding verbal descriptions of fuzzy sets representing fuzzy values. The other is to get linguistic descriptions of fuzzy numbers representing fuzzy uncertainties. The function of the linguistic approximation routine is translating a fuzzy set or a fuzzy number into natural language after the system makes a conclusion.

References

[1] BEZDEK J., Fuzzy models — what are they and why, *IEEE Trans. on fuzzy systems*, Vol. 1, No. 1, pp.1–5, 1993.

[2] BUGARIN A. J., BARRO S., Fuzzy reasoning supported by Petri nets, *IEEE Trans. on fuzzy systems*, Vol. 2, No. 2, pp.135–150, 1994.

[3] CHEN S. M., A new approach to handling fuzzy decisionmaking problems, *IEEE Trans. Syst. Man, Cybern.*, Vol. 18, No. 6, pp.1012–1016, 1988.

[4] CHEN S. M., KE J. S. CHANG J. F., Knowledge representation using fuzzy Petri nets, *IEEE Trans. Syst. Man, Cybern.*, Vol. 2, No. 3, pp.311–319, 1990.

[5] DUBOIS D., PRADE H., Fuzzy sets and systems: theory and applications, *Academic Press*, New York, 1980.

[6] GERKŠIČ M., An extended G–net model for knowledge representation, *The Second German–Slovenian Workshop: 3–D Scene Acquisition, Modeling and Understanding*, in this issue, 1994.

[7] LOONEY C. G., Fuzzy Petri nets for rule — based decisionmaking, *IEEE Trans. Syst. Man, Cybern.*, Vol. 18, No. 1, pp.178–183, 1988.

[8] MURRATA T., Petri nets: properties, analysis and applications, *Proceed. of the IEEE*, Vol. 77, No. 4, pp.541–580, 1989.

[9] PETERSON J. L., Petri net theory and modeling systems, *Prentice Hall*, 1981.

[10] RIBARIĆ S., An application of Petri nets, *Proceed. of Conf. Informatica 78*, pp.2.201–1–2.201–4, 1978.

[11] RIBARIĆ S., Knowledge representation scheme based on Petri net theory, *Int. Journal of Pattern Recognition and Artificial Intelligence*, Vol. 2, No. 4, pp.691–700, 1988.

[12] RIBARIĆ S., An original knowledge representation scheme, *Automatika*, Vol. 32, No. 1–2, pp.13–24, 1991.

[13] ZADEH L. A., Fuzzy sets, *Inform. Contr.*, Vol. 8, pp.338–353, 1965.

[14] ZADEH L. A., Fuzzy logic, *IEEE Comput. Mag.*, Vol. 21, No. 4, pp.83–93, 1988.

[15] ZADEH L. A., Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. Syst. Man, Cybern.*, Vol. SMC–3, No. 1, pp.28–44, 1973.

Contact address:
Mojca Parkelj, Nikola Pavešić
Laboratory for Artificial Perception
Faculty for Electrical Engineering and Computer Science
Tržaška 25, 61000 Ljubljana, Slovenia
University of Ljubljana
e-mail: mojca.parkelj@fer.uni-lj.si

MOJCA PARKELJ (1966) received the B.Sc. degree in electrical engineering in 1992 from the Faculty of Electrical and Computer Engineering at the University of Ljubljana. In 1993 she became a Research Staff Member in the Laboratory of Artificial Perception at the Faculty of Electrical and Computer Engineering in Ljubljana, where she is presently a graduate student and working towards M.Sc. degree in electrical engineering. Her current research interests are around Image Processing, including several scientifi c domains such as Knowledge Representation, Fuzzy Systems, Pattern Recognition and Artificial Intelligence.

NIKOLA PAVEŠIĆ was born in Rijeka, on December 7 1946. After his final exam at the natural science and mathematics department of the II. Gymnasium in Rijeka, he studied at the Faculty for Electrical Engineering, University of Ljubljana. There he received the B.S. degree in electronics, the M.S. degree in automatics and Ph.D. degree in electrical engineering in 1970, 1973 and 1976, respectively. He was the recipient of the Mario Osana Award in 1974, the Vratislav Bedjanič Award in 1976, and the Boris Kidrič Fund Award in 1982. Since 1970 he has been a staff member of the Faculty for Electrical and Computer Engineering in Ljubljana, where he is currently a Full Professor for Systems, Automatics and Cybernetics, Head of the Laboratory for Artificial Perception and Chairman of Department. His research interests include Pattern Recognition and Image Processing, Speech Recognition and Understanding, and Theory of Information. He has authored or co-authored more than 100 papers and 3 books addressing several aspects of the above areas. Dr. Nikola Pavešić is a member of IEEE, the Slovene Association of Electrical Engineers and Technicians (Meritorious Member), The Slovene Pattern Recognition Society (President), and the Slovene Society for Medical and Biological Engineering. He is also a member of editorial boards of several technical journals.