This paper appears in the Proceedings of the World Congress on Neural Networks, vol. 2, pp. 703-708, July 17-21, 1995, Washington D.C.

Design of a Large-Scale Expert System using Fuzzy Logic for Uncertainty-Reasoning[*]

J. Pan and A. C. Kak
1285 E.E. Building
Robot Vision Laboratory
Purdue University
West Lafayette, IN 47907-1285

Abstract

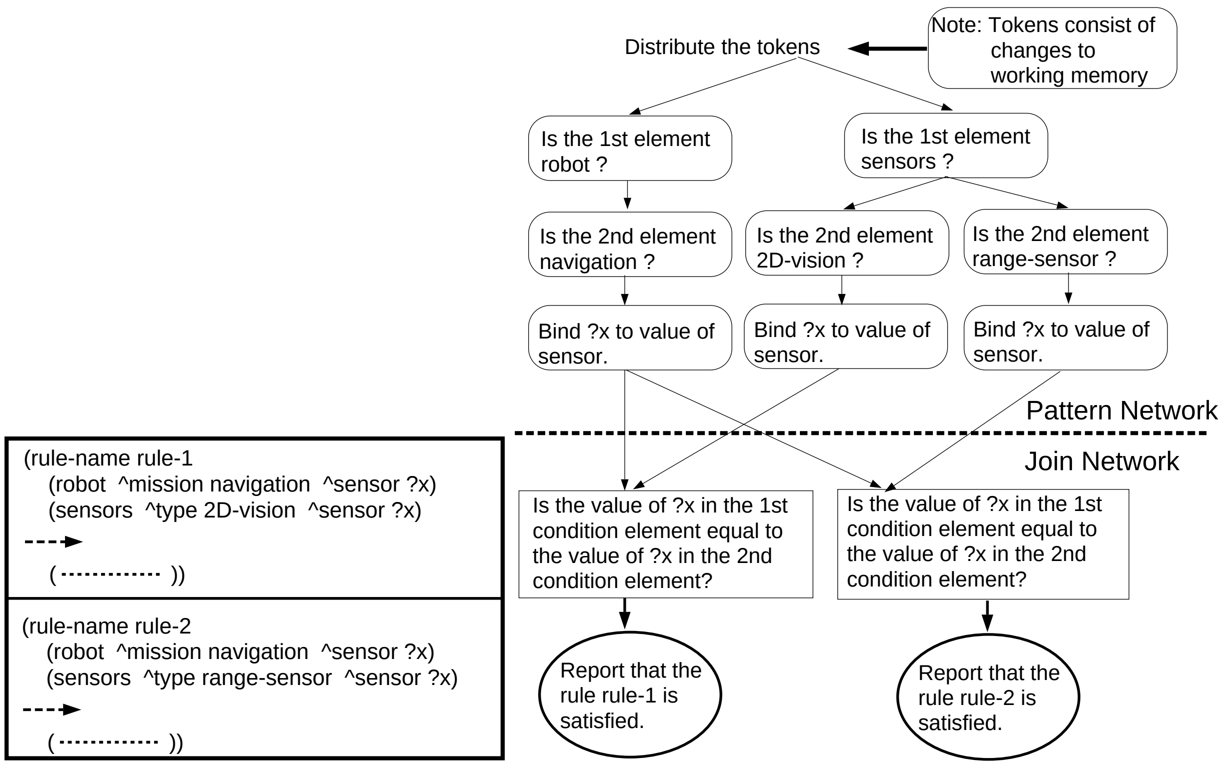
There exist in the literature today many contributions dealing with the incorporation of fuzzy logic in expert systems. But, unfortunately, much of what has been proposed can only be applied to small-scale expert systems, that is when the number of rules is in the dozens as opposed to in the hundreds. Rete networks have been used in the more traditional expert systems to ameliorate the computational burden that would be associated with matching all the rules with all the facts on each cycle of the inference engine. In this paper, we present a more general Rete network that is particularly suitable for reasoning with fuzzy logic. In our new class of Rete networks, before any facts become available, there are the fuzzy membership functions associated with the different terms in the rule-antecedents. Upon the assertion of a fact into the working memory, the pattern matcher "pushes" the fact into the appropriate branches of the network and calculates via a sup-min operation the degree of match between the fact and the rule term. This degree-of-match number is then propagated down the rest of the network in keeping with the rules of the fuzzy logic employed.

1 Introduction

This paper deals with the development of a large-scale expert system using fuzzy logic for uncertainty-reasoning. We must hasten to add that ours is not the first contribution on the subject of how to embed fuzzy logic in an expert system. Actually, many expert systems employing various fuzzy reasoning schemes have been published in the literature [1, 2, 3, 4, 5, 8, 10, 12]. **However, we believe that ours is the first to show how the powerful approach of Rete networks can be used to embed fuzzy logic in an expert system in such a manner that allows the expert system to stay computationally efficient even when the number of rules grows into the hundreds**. When rule-antecedents consist of conjunctive statements, as is usually the case, and when variables are shared between the different statements, it can be shown that simple control structures not employing Rete networks entail exponential complexity in the number of rules on account of the backtracking that is needed to ensure consistent instantiations for the variables in the different statements[6].

We believe that computational burden associated with the use of fuzzy logic in an expert system can be circumvented by generalizing the notion of a Rete network. Rete networks were originally advanced by Forgy [6, 7] to enhance the computational efficiencies of the traditional expert systems, that is expert systems that are based on Boolean logic. The main contribution of this paper is to show how a Rete network can be modified so as to carry out fuzzy inference with computational efficiencies that are comparable to those of the now well-known expert system shells such as OPS5, OPS83, CLIPS, etc. In what follows, we will first review the relevant concepts from fuzzy logic. We will then provide a review of Rete networks, referring the readers to Forgy[6, 7] for a detailed treatment of the subject. That will be followed by a discussion of how Rete networks can be modified to enable them to carry out fuzzy inference.

[*]This work was supported by the Office of Naval Research under Grant ONR N00014-93-1-0142.



Distribute the tokens
Note: Tokens consist of changes to working memory
Is the 1st element robot ?
Is the 1st element sensors ?
Is the 2nd element navigation ?
Is the 2nd element 2D-vision ?
Is the 2nd element range-sensor ?
Bind ?x to value of sensor.
Bind ?x to value of sensor.
Bind ?x to value of sensor.
Pattern Network
Join Network
(rule-name rule-1
(robot ^mission navigation ^sensor ?x)
(sensors ^type 2D-vision ^sensor ?x)
(..............))
(rule-name rule-2
(robot ^mission navigation ^sensor ?x)
(sensors ^type range-sensor ^sensor ?x)
(..............))
Is the value of ?x in the 1st condition element equal to the value of ?x in the 2nd condition element?
Is the value of ?x in the 1st condition element equal to the value of ?x in the 2nd condition element?
Report that the rule rule-1 is satisfied.
Report that the rule rule-2 is satisfied.

Figure 1: Two rules and the corresponding Rete network

2 Fuzzy Logic

To mention quickly the rules of fuzzy logic used in our system, suppose for a linguistic variable U we have two fuzzy sets A and B, with membership functions μ_A and μ_B, respectively, the membership functions of the union and intersection are given by

$$\mu_{A\cup B}(u) = \max[\mu_A(u), \mu_B(u)]$$

$$\mu_{A\cap B}(u) = \min[\mu_A(u), \mu_B(u)]$$

The process of fuzzy inference through rules requires that we associate a fuzzy membership function with each rule. Along the lines discussed in [11] , for the rule "IF A THEN B", a possible fuzzy membership function is

$$\mu_{A\rightarrow B}(u,v) = \min(\mu_A(u), \mu_B(v))$$

As suggested by Zadeh[15], given data A_{data}, fuzzy evidence can be propagated through a rule by using what is usually referred to as a *Compositional Rule of Inference* (CRI). Readers are referred to [9] for propagating evidence through multiple rules with multiple rule antecedents. The CRI we have used can be expressed as

$$\mu_B(v) = \sup_u\{\min_u[\mu_{A_{data}}(u), \mu_{A\rightarrow B}(u,v)]$$

3 Rete Networks

All large-scale expert-system shells of note, such as the CLIPS system from NASA, the commercial systems ART, OPS83 and many others, are founded on the concept of production systems, a concept that was first promulgated by Newell and his co-workers [13] and subsequently made useful for practical problem solving by the seminal work of Forgy on Rete networks [6, 7].

Rete networks have been used in the traditional expert systems to ameliorate the computational burden that would be associated with matching all the rules with all the facts on each cycle of the inference engine. Basically, a Rete network maintains in a network all the matches between the rule-antecedents and the facts and, then, when a new fact becomes available, only the changes are propagated through the network. As demonstrated by Forgy, a Rete network allows two different kinds of redundancies to be eliminated in the workings of the inference engine: It eliminates the temporal redundancies; it also eliminates the structural redundancies, that are redundancies caused by structural similarities between the different statements of a rule-antecedent. A regular Rete network compiles all the rules into a network of feature tests. While rule compilation into feature tests may sound simple, it is quite complicated to program. In Fig.1, we have shown two rule antecedents and the Rete network that would correspond to these two rules. Basically speaking, a Rete network is a tree-structured sorting network containing two parts that are the pattern network and the join network. Once a fact is asserting into the working memory, the fact becomes a token that traverses through the network. The main task of a pattern network is the comparison of the token and the element of a rule antecedent; the main task of a join network is to ensure that the bindings for the variables shared by the rule antecedents are consistent. Note that each node in a pattern network has only one input and each node in a join network has two inputs.

4 Rete Networks for Fuzzy Inference

Our goal in this section is to present a new class of Rete networks that make possible fuzzy inference in large-scale expert systems. In the following, we will show the data structure for fuzzy variables and how the fuzzy evidence of tokens is propagated through the network. For lack of space, we have left out some crucial implementation details here that could help the readers understand better. The interested readers are referred to a very detailed paper[14].

4.1 Data Structure

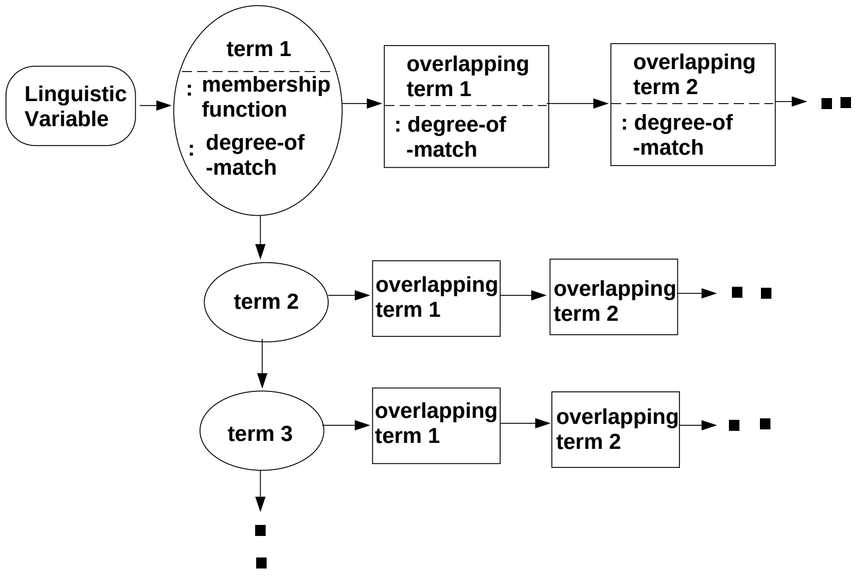


Figure 2: *This data structure facilitates constant-time retrieval of fuzzy terms with overlapping membership functions*

The main differences between this new class of Rete networks and the traditional Rete networks is twofold. We associate a fuzzy membership function which can be piecewise linear with every symbolic variable. The membership function can be changed either as a result of new data in the working memory or as a consequence of the reasoning process. In addition, unlike the case with crisp variables in traditional expert systems, a fact will, in general, enable a rule for firing despite the different symbols appearing in the fact and the rule antecedent. Suppose, we have a fact like (speed fast). This fact will be able to match with the antecedent of a rule like "IF (speed medium) THEN (brake medium)" if there is an overlap between the membership functions for the terms *fast* and *medium* for the linguistic variable *speed*. Fig.2 shows the data structure we use for linguistic variables; this data structure facilitates quick retrieval of all fuzzy terms whose membership functions overlap with a given term. Note especially the dynamic nature of the data structure, meaning that the data structure can accommodate any number of terms for a linguistic variable. The pointers of the overlapping terms are employed for the purpose of matching the different symbols in the token and the rule antecedent during the matching process of the inference engine. It is important to realize that the most recent fuzzy membership functions are stored in the data structures of fuzzy variables. These membership functions are updated at the end of every cycle of the inference engine. Defuzzification of a linguistic variable is achieved simply by accessing the membership functions of fuzzy terms associated with the linguistic variable and by using the approach of "center of area". We are now ready to describe how to propagate fuzzy evidence through the network.

4.2 Propagating Fuzzy Evidence Through Pattern Network

Given a fact like (X is A) and a rule antecedent like (X is A'), if the membership functions of A and A' have an overlap, the match occurs and the value of "degree_of_match" is computed and stored in the corresponding node of the pattern network to indicate the extent to which the fact (X is A) matches the rule-antecedent (X is A'); this degree_of_match then becomes a part of the token that is transmitted down-stream in conjunction with the fact. The precise syntax of this token is

(+(fact-ID), degree_of_match)

Consider, for example, the following patterns which are the antecedents of different rules. Additionally, assume that the fact in the working memory is (robot d̂istance near ŝpeed fast).

Pattern1: (robot d̂istance close ŝpeed fast)
Pattern2: (robot d̂istance far ŝpeed medium)
Pattern3: (robot d̂istance very-far ŝpeed slow)
Pattern4: (robot d̂istance far ŝpeed slow)

In Fig.3, we show four terms, *close, near, far* and *very-far*, associated with the linguistic variable *distance*, and three terms , *slow, medium* and *fast*, associated with the linguistic variable *speed*. Obviously, the overlapping terms for the term *near* contains *close* and *far*. The degree-of-match between *near*, on the one hand and its overlapping terms *close* and *far* is computed by the sup-min operator, as mentioned earlier. Assume the fact *(robot d̂istance near ŝpeed fast)* is asserted into the working memory. When the inference engine tries to match this fact with, say, Pattern1, by accessing the data structure of the term *near*, the match is declared to be successful and a token created to that effect. As shown in Fig.4, since *very-far* is not the overlapping term of *near*, there is no token produced by this branch of the net. The reader should also note that as the token traverses through each node, the transmitted value of degree_of_match is least of the value received by the node and the value calculated at the node. This corresponds to the intersection operation discussed in section 2.

4.3 Propagating Fuzzy Evidence Through Join Network

To illustrate the nature of the join nodes in the network and the fuzzy inference operations that take place in these nodes, consider the following three-rule example and the facts *(robot d̂istance far)* and *(mission ĝoal find-object ŝpeed medium)*:

```
(rule-name rule1
    (robot (d̂istance near) )
    (mission (ĝoal find-object) (ŝpeed fast))
 ⇒ . . .)
(rule-name rule2
     (robot (d̂istance near) )
     (mission (ĝoal find-object) (ŝpeed slow))
  ⇒ . . .)
(rule-name rule3
     (robot (d̂istance near) )
     (mission (ĝoal find-object) (ŝpeed slow))
     (battery high)
  ⇒ . . .)
```

The section of a Rete network that would correspond to these rule antecedents is shown in Fig.5. As shown in the figure, each node for the "end of pattern" is equivalent to one rule antecedent. There is a join node where the condition-element branches come together for each rule and the output of the join node tells us whether or not a rule can be fired. For each node in the join network, it has exactly two inputs which come from either the "end of pattern" or another node of join network. Since there is a conjunctive relationship for the antecedents of a rule, the degree_of_match to be sent by a join node is set to be the minimum of the degree_of_match values for both inputs at each of the join node.

5 Conclusion

We have shown how the fundamental notion of Rete networks can be extended very naturally to fuzzy systems in such a way that the computational efficiencies afforded by such networks are preserved. Although many fuzzy expert systems are now available from commercial vendors or from laboratories, we believe that most of those could not be used when the number of rules is in the hundreds. In this paper, we have shown that if the design of a fuzzy expert system is founded on the concept of Rete networks, the result is a system whose computational efficiency is comparable to that of the more traditional systems. We believe that our contribution here has elucidated the idea of a new class of Rete networks for the design of a large-scale fuzzy expert system.

References

[1] K. P. Adlassnig, "Fuzzy set theory in medical diagnosis," IEEE Trans. Systems, Man and Cybernetics, Vol. 16, pp. 260-265, 1986.

[2] T. Bilgic and I. B. Turksen, "Effective search methods for pattern matching inferencing using specific similarity measures," 1992 IEEE Int'l Conf. on Fuzzy Systems, pp. 161-167.

[3] P. Bonissone, S. Gans and S. Decker, "RUM: A layered architecture for reasoning with uncertainty," in 10th Int'l Joint Conf. on Artificial Intelligence, pp. 891-898, 1987.

[4] J. Buckley and D. Tucker, "Second generation fuzzy expert system," Fuzzy Sets and Systems, Vol. 31, pp. 271-284, 1989.

[5] H. Farreny, H. Prade and E. Wyss, "Approximate reasoning in a rule-based expert system using possibility theory: a case study," Proc. 10th World Computer Congress(IFIP), pp. 407-413, 1986.

[6] C. L. Forgy, "On the efficient implementation of production systems," Ph.D. thesis, Department of Computer Science, CMU, Feb.,1979.

[7] C. L. Forgy, "Rete: A fast algorithm for the many pattern/many object pattern match problem," AI, Vol. 19, pp. 17-37, 1982.

[8] I. Graham, "Fuzzy logic in commercial expert systems - results and prospects", Fuzzy Sets and Systems, pp. 451-472, 1991.

[9] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller – Part 2," IEEE Transaction on Systems, Man and Cybernetics, Vol. 20, pp. 419-435. 1990.

[10] K. S. Leung, W. S. Wong and W. Lam, "Applications of a novel fuzzy expert system shell," Expert Systems: The Int'l J. Knowledge Engineering, Vol. 6, 1989, pp. 2-10.

[11] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis", IEEE Transaction on Computers, Vol. 26, No. 12, pp. 1182-1191, 1977.

[12] M. Mizumoto and H. Zimmermann, "Comparison of fuzzy reasoning methods," Fuzzy Sets and Systems, Vol. 8, pp. 253-283, 1982.

[13] Allen Newell and Herbert A. Simon, Human Problem Solving, Prentice-Hall, 1972.

[14] J. Pan, "Design of a Large-Scale Expert System using Fuzzy Logic for Uncertainty-Reasoning and its application to Vision-Based Mobile Robot Navigation," Preliminary Report, Department of Electrical Engineering, Purdue University, 1994.

[15] L. A. Zadeh, "The role of fuzzy logic in the management of uncertainty in expert system," Fuzzy Sets and Systems 11, North-Holland, pp. 199-227,1983.

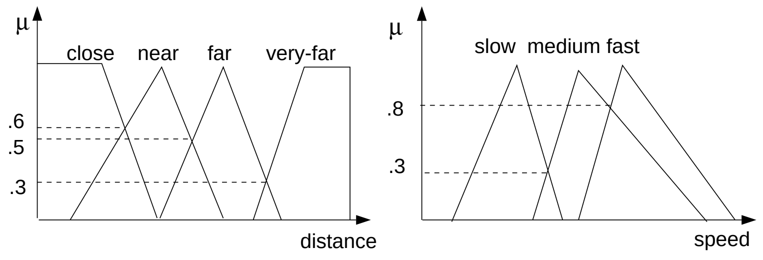


Figure 3: *Membership Functions for the terms of distance and speed.*

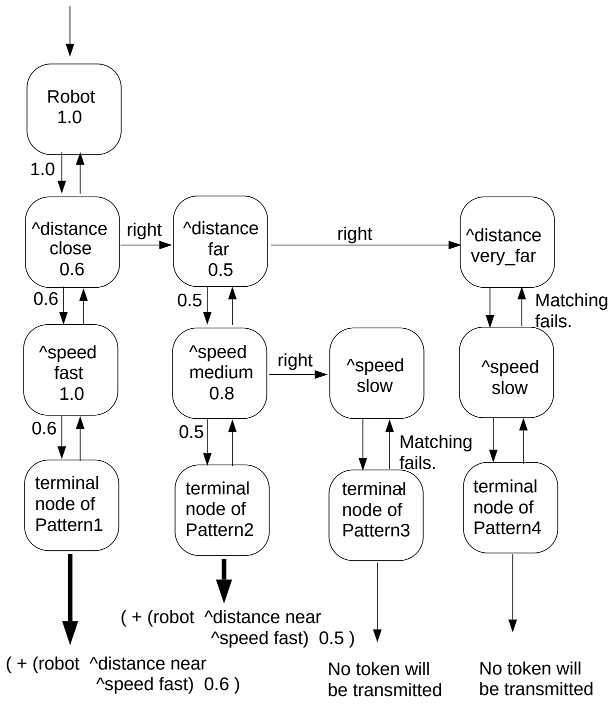

Robot 1.0
1.0
^distance close 0.6
right
^distance far 0.5
right
^distance very_far
0.6
0.5
Matching fails.
^speed fast 1.0
^speed medium 0.8
right
^speed slow
^speed slow
0.6
0.5
Matching fails.
terminal node of Pattern1
terminal node of Pattern2
terminal node of Pattern3
terminal node of Pattern4
(+ (robot ^distance near ^speed fast) 0.5)
(+ (robot ^distance near ^speed fast) 0.6)
No token will be transmitted
No token will be transmitted

Figure 4: *A typical example for propagating fuzzy evidence through pattern network*

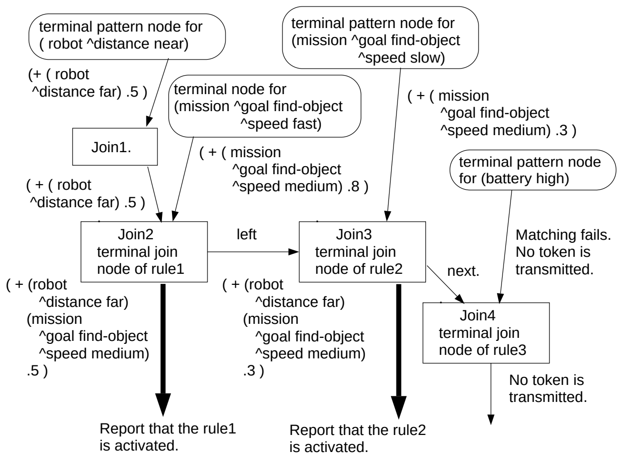

terminal pattern node for (robot ^distance near)
terminal pattern node for (mission ^goal find-object ^speed slow)
(+ (robot ^distance far) .5)
terminal node for (mission ^goal find-object ^speed fast)
(+ (mission ^goal find-object ^speed medium) .3)
Join1.
(+ (mission ^goal find-object ^speed medium) .8)
terminal pattern node for (battery high)
(+ (robot ^distance far) .5)
Join2 terminal join node of rule1
left
Join3 terminal join node of rule2
Matching fails. No token is transmitted.
next.
(+ (robot ^distance far) (mission ^goal find-object ^speed medium) .5)
(+ (robot ^distance far) (mission ^goal find-object ^speed medium) .3)
Join4 terminal join node of rule3
No token is transmitted.
Report that the rule1 is activated.
Report that the rule2 is activated.

Figure 5: *A typical example for propagating fuzzy evidence through join network*